Zhao, J.; Özgen, I.; Liang, D.; Hinkelmann, R.

Improved multislope MUSCL reconstruction on unstructured grids for shallow water equations

Improved multislope MUSCL reconstruction on unstructured grid for shallow water equations

J. Zhao[1*], I.Özgen[1,3], D. Liang[2] and R. Hinkelmann[1]

[1]*Chair of Water Resources Management and Modeling of Hydrosystems, Technische Universität Berlin, Germany*
[2]*Department of Engineering, University of Cambridge, Cambridge, UK*
[3]*Computational Disaster Mitigation and Reduction Research Unit, RIKEN Advanced Institute for Computational Science, Kobe, Japan*

SUMMARY

In shallow water flow and transport modeling, the monotone upstream-centered scheme for conservation laws (MUSCL) is widely used to extend the original Godunov scheme to second-order accuracy. The most important step in MUSCL-type schemes is the MUSCL reconstruction, which calculate extrapolates the values of independent variables from the cell center to the edge. The monotonicity of the scheme is preserved with the help of slope limiters that prevent the occurrence of new extrema during the reconstruction. On structured grids, the calculation of the slope is straightforward and usually based on a two-point stencil that uses the cell centers of the neighbor cell and the so-called far-neighbor cell of the edge under consideration. On unstructured grids, the correct choice for the upwind slope becomes non-trivial. In this work, two novel TVD schemes are developed based on different techniques for calculating the upwind slope and downwind slope. An additional treatment that stabilizes the scheme is discussed. The proposed techniques are compared to two existing MUSCL reconstruction techniques and a detailed discussion of the results is given. It is shown that the proposed MUSCL reconstruction schemes obtain more accurate results with less numerical diffusion and higher efficiency.

KEY WORDS: unstructured mesh; total variation diminishing; slope limiters; finite-volume method; MUSCL scheme; shallow water

1. INTRODUCTION

The monotone upstream-centered scheme for conservation laws (MUSCL) [1] is a well-known approach to achieve higher order accuracy by data reconstruction. An overview of MUSCL-type high-order methods can be found in, e.g. [2, Ch. 13.4, pp. 426–440]. MUSCL-type schemes are essentially an extension of the original Godunov scheme [3]: (1) define the Riemann problems at cell interfaces using cell averages, (2) solve the Riemann problems to get the numerical flux,

*Correspondence to: J. Zhao at Chair of Water Resources Management and Modeling of Hydrosystems, Technische Universität Berlin, secr. TIB1-B14, Gustav-Meyer-Allee 25, 13355 Berlin, Germany. E-mail: jiaheng.zhao@wahyd.tu-berlin.de

(3) update the cell averages by summing up flux and source terms [4]. In contrast, MUSCL-type schemes replace cell averages by piecewise linear functions. Thus, step 1 of the Godunov scheme is replaced by: (1) extrapolate the cell averages (linearly) to cell interfaces in defining the Riemann problems (MUSCL reconstruction step). In order to avoid spurious oscillations, the slope of the extrapolation is limited by so-called slope limiter functions [5]. Many slope limiter functions have been derived in the literature and an overview can be found in [6] and in classical textbooks such as [2, 7].

For high-order schemes to produce physical results, they have to be monotone, otherwise spurious oscillations may occur. Monotonicity of a numerical scheme can be deduced from a property called total variation that is defined as the summation of differences between every two neighboring states over the whole domain at a fixed time. If the total variation does not increase in time, the scheme is said to be total variation diminishing (TVD) and the monotonicity of the scheme is ensured [2]. Slope limiters limit the slope of the MUSCL reconstruction such that it is ensured that no new extrema are created at the cell interfaces. As will be discussed in the following, the limiter function depends on the slope of the upwind and downwind direction of the cell centers. The calculation of these slopes has significant influence on the accuracy of the scheme. Early TVD schemes were derived on structured grids and when applied directly to unstructured meshes give unsatisfactory results. The reason is that on structured grids, the slope calculation on the left and right side is very straight-forward as the direct neighbor cell and the so-called far-neighbor cell (the neighbor of the direct neighbor cell) can be used directly. In addition, the ratio of these slopes is a good indicator for stability because all points that contribute to the ratio are equidistant. In contrast, on unstructured meshes, the calculation of the points that contribute to the calculation of the slopes is not straight-forward. Hence, the calculation of the slopes itself poses a challenge. TVD MUSCL reconstruction techniques for unstructured meshes can be divided into monoslope and multislope methods [8]. The monoslope method that is initially presented in [9] calculates a single vector of slope for the entire cell based on the three direct neighbors of the cell. In contrast, the multislope method calculates a slope for each edge based on a three-point stencil. The challenge of applying the multislope method to unstructured grids is that the determination of the points of the stencil is non-trivial. In literature, several multislope methods for unstructured grids can be found. For example, [10] calculates a local slope based on a two-point stencil without considering the far-neighbor cell. In [11], a three-point stencil is used but instead of the far-neighbor cell, a “virtual” node is included. Motivated by these approaches, in [6] a multislope method that calculates individual weights for slopes depending on the distance of the cell centers to the cell interface is derived. In [12] slopes are calculated based on points that are located on a line normal to the cell interface. Based on the multislope method by Buffard and Clain [8], Hou *et al.* [13] proposed a more straightforward vector based multislope method, which obtains the robustness and accuracy together.

In this work, two novel TVD MUSCL reconstruction techniques are investigated based on an additional condition for the TVD scheme, and a treatment for limiting the velocity at wet and dry interfaces is proposed for avoiding the instability caused by MUSCL schemes. The proposed schemes are compared with the multislope methods by [8] and [13]. The schemes are compared in five computational test cases: (1) Thacker’s planar rotation benchmark, (2) a steady-state oblique jump, (3) a radial dam-break, (4) a two-dimensional Riemann problem, and (5) a Tsunami wave

around a canonical island. In these benchmarks, the accuracy of the TVD method as well as their computational cost is compared.

2. GOVERNING EQUATIONS AND NUMERICAL MODEL

2.1. Shallow water equations

The two-dimensional shallow water equations (SWEs) can be written in vector form as

$$\frac{\partial \mathbf{q}}{\partial t} + \frac{\partial \mathbf{f}}{\partial x} + \frac{\partial \mathbf{g}}{\partial y} = \mathbf{s} \tag{1}$$

with vectors defined as

$$\mathbf{q} = \begin{bmatrix} h \\ q_x \\ q_y \end{bmatrix}, \quad \mathbf{f} = \begin{bmatrix} q_x \\ uq_x + gh^2/2 \\ uq_y \end{bmatrix}, \quad \mathbf{g} = \begin{bmatrix} q_y \\ vq_x \\ vq_y + gh^2/2 \end{bmatrix}, \tag{2}$$

$$\mathbf{s} = \begin{bmatrix} 0 \\ -gh\frac{\partial z}{\partial x} - c_f u\sqrt{u^2+v^2} \\ -gh\frac{\partial z}{\partial y} - c_f v\sqrt{u^2+v^2} \end{bmatrix}, \tag{3}$$

where t is time, x and y are the Cartesian coordinates, $\mathbf{q}$ represents the variable vector consisting of h, q_x and q_y that denote water depth, unit-width discharges in x- and y- direction, respectively. u, v are defined as depth-averaged velocities in x- and y-direction, respectively; $\mathbf{f}$ and $\mathbf{g}$ are the flux vectors in x- and y-direction, respectively; $\mathbf{s}$ is the source term that includes bed slope and friction contributions, z is the bed elevation and c_f is the bed roughness coefficient. Here, viscous, diffusive and turbulent flux terms are neglected.

2.2. Finite volume discretization of SWEs on unstructured grids

The SWEs in Eq. (1) can be written in the integral form as

$$\int_\Omega \frac{\partial \mathbf{q}}{\partial t} d\Omega + \int_\Omega \left(\frac{\partial \mathbf{f}}{\partial x} + \frac{\partial \mathbf{g}}{\partial y} \right) d\Omega = \int_\Omega \mathbf{s} d\Omega \tag{4}$$

where Ω is an arbitrary control volume (CV). Applying the Green-Gauß theorem and replacing the boundary integral with a sum over all edges, Eq. (4) becomes

$$\int_\Omega \frac{\partial \mathbf{q}}{\partial t} d\Omega + \sum_{k=1}^{m} \mathbf{F} \cdot \mathbf{n}_k l_k = \int_\Omega \mathbf{s} d\Omega \tag{5}$$

wherein m is the number of edges, and $\mathbf{n} = (n_x, n_y)^T$, is the unit normal vector pointing in the outward normal direction of the boundary edge, l is the length of the edge, $\mathbf{F} \cdot \mathbf{n}$ is the flux vector normal to the boundary and can be written as

$$\mathbf{F}\cdot\mathbf{n} = (\mathbf{f}\mathrm{n_x} + \mathbf{g}\mathrm{n_y}) = \begin{bmatrix} q_x n_x + q_y n_y \\ (uq_x + gh^2/2)n_x + vq_y n_y \\ uq_x n_x + (vq_y + gh^2/2)n_y \end{bmatrix}. \tag{6}$$

The value of $\mathbf{q}$ in cell i is updated using the two-stage explicit Runge-Kutta scheme [14–16], where the value at the next time level in cell i, $\mathbf{q}_i^{n+1}$, is updated by

$$\mathbf{q}_i^{n+1} = \frac{1}{2}\left\{\mathbf{q}_i^n + \kappa\left[\kappa\left(\mathbf{q}_i^n\right)\right]\right\} \tag{7}$$

with

$$\kappa(\mathbf{q}_i^n) = \mathbf{q}_i^n + \frac{\Delta t^n}{\Omega}\left[\int_\Omega \mathbf{s}^{n+1} d\Omega - \sum_{k=1}^{m} \mathbf{F}(\mathbf{q}_i^n)_k \cdot \mathbf{n}_k l_k\right], \tag{8}$$

where $\mathbf{s}^{n+1}$ is friction source term and discretized in a splitting point implicit way [17], the slope source term is calculated based on the slope flux calculation method from [18], which is added into flux term $\mathbf{F}(\mathbf{q}_i^n)$, κ is a function to represent the updating process to a new time level in the considered cell. Δt^n is the time step at the nth time level. For this work, the Courant-Friedrichs-Lewy condition is used here for maintaining the stability,

$$\Delta t = \mathrm{CFL}\min\left(\frac{R_1}{\sqrt{u_1^2 + v_1^2} + \sqrt{gh_1}}, ..., \frac{R_n}{\sqrt{u_n^2 + v_n^2} + \sqrt{gh_n}}\right) \tag{9}$$

where R_n is the minimum distance from the cell center to the edge, CFL is the Courant-Friedrichs-Lewy number. For explicit time marching algorithms $\mathrm{CFL} \in (0, 1]$. In this work, $\mathrm{CFL} = 0.5$ is adopted.

2.3. MUSCL reconstruction

In order to obtain a second order accurate numerical scheme, Godunov's theorem [3] can be circumvented by reconstructing the cell-averaged values linearly using the MUSCL reconstruction [1]. The MUSCL reconstruction is applied successfully for many physical problems, e.g. [2, 16, 19–23]. The reconstruction from cell center i to the cell interface $(i, i+1)$, hereinafter also denoted with the subscript $i + 1/2$, is calculated as

$$q_{i+1/2} = q_i + \Delta x_{i,i+1/2}\psi(r)\frac{q_{i+1} - q_i}{\Delta x_{i,i+1/2} + \Delta x_{i+1/2,i+1}} \tag{10}$$

as shown in Fig. 1, $\Delta x_{i,i+1/2}$ and $\Delta x_{i+1/2,i+1}$ are the distances from the cell center i to the edge and the cell center $i+1$ to the edge, respectively, ψ is the limiting coefficient of slope, and r is the slope ratio, which will be discussed in the following section.

The MUSCL reconstruction gives values at the left and right cell interface that can be used to construct a Riemann problem. The solution of the Riemann problem then yields the numerical flux in Eq. (6) [2]. In this work, a Harten, Lax, and van Leer Riemann solver with the contact wave restored (HLLC) [24] is used. The positivity preserving hydrostatic reconstruction by [25] is used

to maintain non-negative water depth and correct reconstruction of the Riemann states, and the C-property preserving divergence form of the bed slope source term proposed by Hou *et al.* [18] is used; the source term treatment does not influence the well-balanced property of any of the MUSCL schemes.

3. MULTISLOPE MUSCL RECONSTRUCTION METHODS

In multislope MUSCL reconstruction methods, the slope for the MUSCL reconstruction in Eq. (10) is calculated at each edge individually.

In the original multislope scheme (derived in [26] for uniform grids), the edge value is calculated with a diffusive first order upwind value and an anti-diffusive term as

$$q_{CD} = q_C + \frac{1}{2}\psi(r)(q_D - q_C). \tag{11}$$

Here, r is the ratio of consecutive slopes [27], that can be calculated with a three-point-stencil that consists of two adjacent cells C and D, and the far-neighbor cell U located in the upstream direction. The ratio r becomes:

$$r = \frac{\bigtriangledown q_{up}}{\bigtriangledown q_{down}} \tag{12}$$

It is noted that due to the uniform grid assumption, the ratio $\frac{\Delta x_{i,i+1/2}}{\Delta x_{i,i+1/2}+\Delta x_{i+1/2,i+1}}$ in Eq. (10) is simplified to $\frac{1}{2}$ in Eq. (11).

In literature, two-dimensional (2D) MUSCL schemes on unstructured grids are mainly separated into one-dimensional (1D) gradient methods, e.g. [11], [10], [6], [12], 2D nodal evaluating methods, e.g. [28], [29], and vector manipulation methods, e.g. [13], [8]. In this paper, 2D nodal evaluating methods and vector manipulation methods are compared with regard to efficiency, accuracy and ease of implementation.

3.1. 2D nodal evaluation methods

A straight-forward approach to calculate the MUSCL slope on unstructured grids is to directly apply the classical TVD methods derived for 1D structured grids. Then, the upstream node U of the stencil can be calculated by extrapolating along a certain distance in the upstream direction. Darwish and Moukalled [10] note that the difficulty of implementing MUSCL reconstructions on unstructured grids is determining U. Based on prior work by Bruner and Walters [30], Darwish and Moukalled [10] proposed a MUSCL reconstruction method (Darwish's scheme) where r is defined as

$$r = \frac{2(\bigtriangledown q)_C \cdot \overrightarrow{d}_{C,D}}{q_D - q_C} - 1. \tag{13}$$

Here, $\overrightarrow{d}_{C,D}$ is the distance vector from cell center of C to cell center of D. $(\bigtriangledown q)_C$ is the cell value gradient of cell C. In contrast to the calculation of r on uniform grids given in Eq. (12), Darwish and Moukalled [10] account for non-uniform distance between the cell centers. Instead of determining

an additional point U, the value at node U is interpolated based on the gradient of cell C, as shown in Fig. 2 (left). This is in fact a very local calculation of r. If the gradient in cell C does not represent the overall behavior of the variable, the calculated r differs significantly from a three-point-stencil.

Li and Liao [11] define the stencil for calculating r using the cell centers of two adjacent cells C and D, and construct a virtual node U which is located in the upstream direction on the line connecting C and D, such that all nodes are equidistant, cf. Fig. 2 (left). U_r is the cell center where node U is located in. The value at U is interpolated based on the gradient of the cell containing U ($(\bigtriangledown q)_{U_r}$):

$$q_U = q_{U_r} + \overrightarrow{d}_{U_r,U} \cdot (\bigtriangledown q)_{U_r} \tag{14}$$

and Eq. (12) becomes:

$$r = \frac{q_C - (q_{U_r} + \overrightarrow{r_{U_r,U}} \cdot (\bigtriangledown q)_{U_r})}{q_D - q_C} \tag{15}$$

The method by Li and Liao [11] contains more upstream information and is not as local as Darwish's scheme.

All of these TVD schemes neglect the interface position and the distance of the cell centers to the interface. Hou *et al.* [6] propose a reconstruction (Hou's 1st scheme) that includes the interface position in the interpolation. It can be written as

$$q_{CD} = q_C + \frac{d_{Cf}}{d_{Cf} + d_{Df}} \psi(r_f)(q_D - q_C), \tag{16}$$

$$r = \frac{(q_D - q_C)/d_{CD}}{(q_C - q_U)/d_{UC}}. \tag{17}$$

Here, f denotes the interface, d_{UC}, d_{CD}, d_{Cf}, d_{Df} are the distances from node U to C, C to D, C to interface and D to interface, respectively, cf. Fig. 2 (left).

Hou *et al.* [12] note that in most reconstruction techniques, the value at node U is interpolated from the cell center using the gradient of the variable in the cell. Thus, the cell gradient has to be calculated firstly. It is then argued that in advection dominated flows, the flux can be splitted in a component normal to the interface and a component tangential to the interface. The tangential component is transported by the normal component as a passive scalar. Therefore, instead of connecting the cell centers of the adjacent cell and locate the upstream node U on this line, Hou *et al.* [12] suggest to draw a line that goes normal to the interface, through the center of the interface and locate all three nodes of the stencil on this line, cf. Fig. 2 (right). C, D and U are projections of the cell centers C_r, D_r and U_r, respectively. U_r is determined as the nearest cell center around the vertex P_1 of cell C to the line. The values at the points C, D and U are not interpolated but shifted directly from the cell centers, i.e. the value at node U is equal to the value at node U_r. Then, Eqs. (16) and (17) are used to reconstruct the values at interface.

3.2. Vector manipulation methods

Buffard and Clain [8] developed an approach to construct the upwind and downwind slope of edges in the considered cells by manipulating the geometric directional unit vector. Based on this

approach, Hou *et al.* [13] developed a more straightforward scheme for the calculation of the upwind and downwind slopes.

As shown in Fig. 3, dimensional unit vectors are calculated as

$$\overrightarrow{r_k} = \frac{\overrightarrow{iM_k}}{|\overrightarrow{iM_k}|} \tag{18}$$

$$\overrightarrow{t_k} = \frac{\overrightarrow{ij_k}}{|\overrightarrow{ij_k}|}. \tag{19}$$

M is the center of the edge, $\overrightarrow{r_k}$ and $\overrightarrow{t_k}$ are the unit directional vectors from considered cell center to the edge center and neighbor cell center respectively. It also can easily be shown that the opposite direction of $\overrightarrow{r_k}$ will pass by the node coordinate which belongs to the considered cell but not the vertex of the edge.

The slope along $\overrightarrow{t_k}$ can be calculated as

$$\bigtriangledown q_{ij} = \frac{q_{j_k} - q_i}{|ij_k|}. \tag{20}$$

By following the approach of Buffard and Clain [8], the slopes along the line connecting the cell center with the edge center in the upwind and downwind direction can be thought as the slopes from the far node N_m, with $m = 1, 2, 3$, to the cell center i and from the cell center i to the considered edge center M, respectively, i.e the slopes along $-\overrightarrow{r_k}$ and $\overrightarrow{r_k}$ in Fig. 3, respectively.

In the scheme by Hou *et al.* [13], the vectors $\overrightarrow{r_k}$ and $-\overrightarrow{r_k}$ are obtained by solving a set of linear equations, that can be obtained by geometric considerations as

$$\overrightarrow{r_1} = \alpha_1 \overrightarrow{t_1} + \alpha_2 \overrightarrow{t_3} \tag{21}$$

$$-\overrightarrow{r_1} = \beta_1 \overrightarrow{t_2} + \beta_2 \overrightarrow{t_3}. \tag{22}$$

Here, $\alpha_{1,2}$ and $\beta_{1,2}$ are coefficients for linear construction and can be calculated from Eqs. 21 and 22. The slopes along $\overrightarrow{t_k}$ are obtained by Eq. 20. The gradients along $\overrightarrow{t_k}$ and $\overrightarrow{r_k}$ are independent from each other and $(\mathrm{grad} \cdot \overrightarrow{t})_{i,j} = \bigtriangledown q_{i,j}$, so that slopes at the upwind and downwind direction can be calculated as

$$\bigtriangledown q_{iM} = \alpha_1 \bigtriangledown q_{i,j_1} + \alpha_2 \bigtriangledown q_{i,j_2} \tag{23}$$

$$\bigtriangledown q_{N_1,i} = -(\beta_1 \bigtriangledown q_{i,j_2} + \beta_2 \bigtriangledown q_{i,j_3}). \tag{24}$$

Values at the cell edge are defined as q_M^i and q_M^o, where the superscripts i and o denote that the variable is defined at the inside or at the outside of the cell under consideration, respectively. It is noted, that the variables at the outside of the cell under consideration equal the inside variables of the neighbor cell. q_M^i can be calculated as

$$q_M^i = q_i + |iM_k| \psi(\bigtriangledown q_{iM_k}, \bigtriangledown q_{N_j i}) \tag{25}$$

where k is the local index of the edge, and j is the local index of the node along the opposite direction of the considered edge. ψ is the limiter function as defined in the previous section, with the difference being that the parameters here are using the upwind and downwind slopes instead of the slope ratio r.

3.3. Methods of improving the TVD property

One aspect that to the authors' knowledge is not discussed in the literature is a special case of MUSCL reconstruction that violates monotonicity without creating new extrema. Consider the case illustrated in Fig. 4. If $r_{(C,D)} > 1.0$ and $r_{(D,C)} > 1.0$, where $r_{(C,D)}$ is the slope ratio on the left side of the edge and $r_{(D,C)}$ is the slope ratio on the right side of the edge, the reconstructed values give $q_{eL} > q_{eR}$ even though $q_C < q_D$. The classical TVD limiter does not prevent this case because no new extremum is created. However, the numerical flux based on q_{eL} and q_{eR} in this case is physically not feasible.

In this work, it is ensured that the following condition is satisfied, so that the case discussed above is always prevented:

$$(q_{i+1} - q_i)(q_{i+1,i} - q_{i,i+1}) \geq 0.0 \tag{26}$$

This condition preserves the monotonicity for both cells and edges. The derivation of Eq. (26) is briefly presented in the following.

The TVD condition for the one-dimensional case is given in [5] as:

$$q_i^{n+1} = q_i^n + C_{+,(i,i+1)}\Delta q_{i,i+1}^n - C_{-,(i-1,i)}\Delta q_{i-1,i}^n \tag{27}$$

Here, C is a variable-dependent coefficient, subscript '$+$' and '$-$' mean the flux flow into and out of cell i respectively, $\Delta q_{i,i+1}^n = q_{i+1}^n - q_i^n$. The sufficient TVD conditions are expressed in a series of inequalities:

$$C_{+,(i,i+1)} \geq 0, \quad C_{-,(i,i+1)} \geq 0, \quad C_{+,(i,i+1)} + C_{-,(i,i+1)} \leq 1 \tag{28}$$

A TVD scheme needs to satisfy the conditions given in Eqs. (27 and 28). For one dimensional grids, the variable at time step $n+1$ in the cell i can be written as

$$q_i^{n+1} = q_i^n + C_{+,(i,i+1)}(q_{i+1} - q_i) - C_{-,(i-1,i)}(q_i - q_{i-1}), \tag{29}$$

and the reconstructed variables read

$$q_{i+1,i} = q_{i+1} + \Upsilon_{(i+1,i)}(q_i - q_{i+1}) \tag{30}$$

$$q_{i,i+1} = q_i + \Upsilon_{(i,i+1)}(q_{i+1} - q_i) \tag{31}$$

$$q_{i,i-1} = q_i + \Upsilon_{(i,i-1)}(q_{i-1} - q_i) \tag{32}$$

$$q_{i-1,i} = q_{i-1} + \Upsilon_{(i-1,i)}(q_i - q_{i-1}). \tag{33}$$

Υ can be seen as the distance multiplied with the slope limiter. Then, it is easy to get

$$q_{i+1} - q_i = \frac{q_{i+1,i} - q_{i,i+1}}{1.0 - \Upsilon_{(i+1,i)} - \Upsilon_{(i,i+1)}} \tag{34}$$

$$q_i - q_{i-1} = \frac{q_{i,i-1} - q_{i-1,i}}{1.0 - \Upsilon_{(i,i-1)} - \Upsilon_{(i-1,i)}} \tag{35}$$

so that the Eq. (29) can be rewritten as:

$$q_i^{n+1} = q_i^n + \frac{C_{+,(i,i+1)}}{1.0 - \Upsilon_{(i+1,i)} - \Upsilon_{(i,i+1)}}(q_{i+1,i}^n - q_{i,i+1}^n) - \frac{C_{-,(i-1,i)}}{1.0 - \Upsilon_{(i,i-1)} - \Upsilon_{(i-1,i)}}(q_{i,i-1}^n - q_{i-1,i}^n), \tag{36}$$

$$\begin{aligned}
\Delta q_{i,i+1}^{n+1} &= q_{i+1}^{n+1} - q_i^{n+1} \\
&= q_{i+1}^n + \frac{C_{+,(i+1,i+2)}}{1.0 - \Upsilon_{(i+2,i+1)} - \Upsilon_{(i+1,i+2)}}(q_{i+2,i+1}^n - q_{i+1,i+2}^n) - \frac{C_{-,(i,i+1)}}{1.0 - \Upsilon_{(i+1,i)} - \Upsilon_{(i,i+1)}}(q_{i+1,i}^n - q_{i,i+1}^n) \\
&- (q_i^n + \frac{C_{+,(i,i+1)}}{1.0 - \Upsilon_{(i+1,i)} - \Upsilon_{(i,i+1)}}(q_{i+1,i}^n - q_{i,i+1}^n) - \frac{C_{-,(i-1,i)}}{1.0 - \Upsilon_{(i,i-1)} - \Upsilon_{(i-1,i)}}(q_{i,i-1}^n - q_{i-1,i}^n)) \\
&= \frac{q_{i+1,i}^n - q_{i,i+1}^n}{1.0 - \Upsilon_{(i+1,i)} - \Upsilon_{(i,i+1)}} + \frac{C_{+,(i+1,i+2)}}{1.0 - \Upsilon_{(i+2,i+1)} - \Upsilon_{(i+1,i+2)}}(q_{i+2,i+1}^n - q_{i+1,i+2}^n) \\
&- \frac{C_{-,(i,i+1)}}{1.0 - \Upsilon_{(i+1,i)} - \Upsilon_{(i,i+1)}}(q_{i+1,i}^n - q_{i,i+1}^n) - (\frac{C_{+,(i,i+1)}}{1.0 - \Upsilon_{(i+1,i)} - \Upsilon_{(i,i+1)}}(q_{i+1,i}^n - q_{i,i+1}^n) \\
&- \frac{C_{-,(i-1,i)}}{1.0 - \Upsilon_{(i,i-1)} - \Upsilon_{(i-1,i)}}(q_{i,i-1}^n - q_{i-1,i}^n)) \\
&= \frac{1.0 - C_{-,(i,i+1)} - C_{+,(i,i+1)}}{1.0 - \Upsilon_{(i+1,i)} - \Upsilon_{(i,i+1)}}(q_{i+1}^n - q_{i,i+1}^n) + \frac{C_{+,(i+1,i+2)}}{1.0 - \Upsilon_{(i+2,i+1)} - \Upsilon_{(i+1,i+2)}}(q_{i+2,i+1}^n - q_{i+1,i+2}^n) \\
&+ \frac{C_{-,(i-1,i)}}{1.0 - \Upsilon_{(i,i-1)} - \Upsilon_{(i-1,i)}}(q_{i,i-1}^n - q_{i-1,i}^n).
\end{aligned} \tag{37}$$

For TVD schemes, it is not difficult to get $1.0 - C_{-,(i,i+1)} - C_{+,(i,i+1)} \geq 0, C_{+,(i+1,i+2)} \geq 0, C_{-,(i-1,i)} \geq 0$. If Eq. (37) satisfies the TVD property, the coefficients should be non-negative, which gives that $1.0 - \Upsilon_{(i,i-1)} - \Upsilon_{(i-1,i)}, 1.0 - \Upsilon_{(i+1,i)} - \Upsilon_{(i,i+1)}, 1.0 - \Upsilon_{(i+2,i+1)} - \Upsilon_{(i+1,i+2)}$ should all be positive values and thus

$$\begin{aligned}
|\Delta q_{i,i+1}^{n+1}| \leq \frac{1.0 - C_{-,(i,i+1)} - C_{+,(i,i+1)}}{1.0 - \Upsilon_{(i+1,i)} - \Upsilon_{(i,i+1)}}|q_{i+1}^n - q_{i,i+1}^n| + \frac{C_{+,(i+1,i+2)}}{1.0 - \Upsilon_{(i+2,i+1)} - \Upsilon_{(i+1,i+2)}}|q_{i+2,i+1}^n - q_{i+1,i+2}^n| \\
+ \frac{C_{-,(i-1,i)}}{1.0 - \Upsilon_{(i,i-1)} - \Upsilon_{(i-1,i)}}|q_{i,i-1}^n - q_{i-1,i}^n|.
\end{aligned} \tag{38}$$

Summing up Eq. (38) for $-\infty < i < \infty$ the total variation at time $n+1$ gives

$$TV(q^{n+1}) = \sum_{i=-\infty}^{i=\infty} |\Delta q_{i,i+1}^{n+1}| \leq \sum_{i=-\infty}^{i=\infty} \frac{1.0 - C_{-,(i,i+1)} - C_{+,(i,i+1)}}{1.0 - \Upsilon_{(i+1,i)} - \Upsilon_{(i,i+1)}} |q_{i+1}^n - q_{i,i+1}^n|$$
$$+ \sum_{i=-\infty}^{i=\infty} \frac{C_{+,(i+1,i+2)}}{1.0 - \Upsilon_{(i+2,i+1)} - \Upsilon_{(i+1,i+2)}} |q_{i+2,i+1}^n - q_{i+1,i+2}^n|$$
$$+ \sum_{i=-\infty}^{i=\infty} \frac{C_{-,(i-1,i)}}{1.0 - \Upsilon_{(i,i-1)} - \Upsilon_{(i-1,i)}} |q_{i,i-1}^n - q_{i-1,i}^n|$$
$$= \sum_{i=-\infty}^{i=\infty} |\Delta q_{i,i+1}^n| = TV(q^n). \tag{39}$$

It is noted that Υ depends on the slope limiter Ψ and the position information $\Delta x_{i,i+1/2}/(\Delta x_{i,i+1/2} + \Delta x_{i+1/2,i+1})$, so Eqs. (30) - (36) are also valid for unstructured grids. If $1.0 - \Upsilon_{(i+1,i)} - \Upsilon_{(i,i+1)}$ and $1.0 - \Upsilon_{(i,i-1)} - \Upsilon_{(i-1,i)}$ are zero, which means the reconstructed values along the edge for both neighbors are the same, the absolute value for the value difference is zero and does not influence the TVD property of this scheme, as

$$1.0 - \Upsilon_{(i+1,i)} - \Upsilon_{(i,i+1)} \geq 0, \tag{40}$$

if $q_{i+1} \geq q_i$, Eq. (40) can be rewritten as

$$q_i + \gamma_{i,i+1}(q_{i+1} - q_i) - (q_{i+1} + \gamma_{i+1,i}(q_i - q_{i+1})) \leq 0.0, \tag{41}$$

it can be obtained that if $q_i \leq q_{i+1}$ then $q_{i,i+1} \leq q_{i+1,i}$, and vice versa, it will be very easy to get the relationship for the variables along the edges,

$$(q_{i+1} - q_i)(q_{i+1,i} - q_{i,i+1}) \geq 0.0 \tag{42}$$

which is the condition given by Eq. (26). □

If the TVD condition is satisfied for the reconstruction method, the scheme should fullfill the relationship given in Eq. (28), but if the upwind and downwind neighbor cells are defined as $i-2$ and $i+2$ respectively, Eq. (28) becomes

$$q_i^{n+1} = q_i^n + C_{+,(i,i+2)} \Delta q_{i,i+1}^n - C_{-,(i-2,i)} \Delta q_{i-1,i}^n. \tag{43}$$

The value difference between cell i and $i+1$ on the new time step $n+1$ can be stated as,

$$\Delta q_{i,i+1}^{n+1} = (1 - C_{+,(i,i+2)} - C_{-,(i,i+2)}) \Delta q_{i,i+1}^n + C_{+,(i+1,i+3)} \Delta q_{i+1,i+2}^n + C_{-,i-2,i} \Delta q_{i-1,i}^n. \tag{44}$$

If all the coefficients satisfy the inequalities

$$C_{+,(i,i+2)} \geq 0, \quad C_{-,(i,i+2)} \geq 0, \quad C_{+,(i,i+2)} + C_{-,(i,i+2)} \leq 1, \tag{45}$$

the total variation can be summed up for the $-\infty < i < +\infty$ and the following expression is obtained:

$$TV(q^{n+1}) = \sum_{i=-\infty}^{i=\infty} |\Delta q_{i,i+1}^{n+1}| \leq \sum_{i=-\infty}^{i=\infty} (1.0 - C_{-,(i,i+2)} - C_{+,(i,i+2)})|\Delta q_{i.i+1}^{n}| + \sum_{i=-\infty}^{i=\infty} C_{+,(i+1,i+3)}|\Delta q_{i+1,i+2}^{n}| + \sum_{i=-\infty}^{i=\infty} (C_{-,(i-2,i)})|\Delta q_{i-1,i}^{n}| \quad (46)$$

Changing the indices of the last two terms to $i+2$ and $i-2$, respectively, it is seen that the resulting equation is not TVD anymore:

$$TV(q^{n+1}) = \sum_{i=-\infty}^{i=\infty} |\Delta q_{i,i+1}^{n+1}| \leq \sum_{i=-\infty}^{i=\infty} (1.0 - C_{-,(i,i+2)} + C_{-,(i-1,i+1)})|\Delta q_{i.i+1}^{n}| \neq TV(q^{n}). \quad (47)$$

It can be concluded that the coefficients which get the influence from the $i-2$ and $i+2$ will not lead to a TVD scheme. As shown in Fig. 5, the situation can be thought about as a dam break problem. From the time level t to $t+1$, if $i \pm 2$ are chosen to be the upwind and downwind cells, the slope for the water depth (left) is calculated properly, but the slope for the discharge (right), $i+2$ will yield a wrong interpolation if the cell i is used as the upwind cell instead of $i+1$.

Considering the previous 2D node evaluation methods, only Darwish's scheme satisfies the condition presented in this section. In contrast, the other 2D node evaluation schemes using a certain distance or preferential direction may set $i \pm (n \neq 1)$ as the upwind cell, thus leading to a non-stable scheme. However, Darwish's scheme consists of a very localized stencil and therefore has a lower order of accuracy.

In [28, 29], the stability condition of 2D shallow water equations is extended to an interval from the minimum and maximum value of the cell centered values at both sides of the considered edge and the values of the two vertices of that edge, so that it will allow that the interpolated values of the edge can be the values larger or lesser than the values at the cell centers of both adjacent cells.

The TVD condition for the vector manipulation method is obtained by verifying that the scheme satisfies Eq. (26). As an illustrative example, a Scottish mesh is chosen to calculate the stencil of the reconstruction method, as shown in Fig. 6. α_1, α_2 and β_1, β_2 are calculated to be 1.34164, 0.632456 and 1.26491, 1.34164. Because the vectors $\overrightarrow{r_1}$ and $-\overrightarrow{r_1}$ are located between the vectors, they will be used for constructing the linear system later, it can be shown that all of the values are non-negative numbers, so that the slope for the upwind and downwind is depending on $\overrightarrow{t_j}$. Let $h_D = 1.1\,\mathrm{m}$, $h_E = 0.2\,\mathrm{m}$, then the reconstruction of $h_{D,M} = 1.1\,\mathrm{m}$ remains first order, but for $q_{C,M}$, the slopes for the upwind and downwind in cell C can be calculated as $\bigtriangledown q_{N,C} = 0.7589$, $\bigtriangledown q_{C,M} = 0.7589$, which means there will be a positive value calculated by the limiter function and multiplied with the distance, which may create a local extremum value larger than $q_{D,M}$. If $h_D = 1.0$, the overestimated interpolation also will happen and will lead to the unphysical flux.

4. IMPROVED MULTISLOPE MUSCL RECONSTRUCTION METHODS

For multislope methods on unstructured grids, the three-point-stencil can be considered as a one-dimensional interpolation stencil problem in the local coordinate system of the edge, whereby the main challenge becomes the definition of the upstream node. In this work, it is suggested to use the upwind values interpolated or extrapolated by the surrounding neighbor cell center values. The details are given in the following.

In the following, it is assumed that all the multislope MUSCL methods require that the cell center of the considered cell is located inside the triangle constructed by the cell centers of the surrounding neighbors, cf. Fig. 7.

4.1. Improved 2D node evaluation method

Motivated by the reconstruction method proposed in [12], which is based on the fact that in advection-dominated problems the flux at the interface mainly depends on the variables transport normal to this interface, the following TVD reconstruction is suggested:

1. Draw a line perpendicular to the considered edge, which passes through the center of the edge
2. Find the intersection node U of the normal line and the line connecting the neighboring cell centers which are the outside neighbors of the edges in the considered cell except the considered edge. cf. Fig. 8a.
3. Interpolate (if U is located between E and F) or extrapolate (if U is not located between E and F).
4. Reconstruct the values.

The main novelty of this method is the choice of the upstream node U. The upwind cell of the considered cell for the considered edge is chosen from the neighboring cells of the considered cell. As illustrated in Fig. 8a, these are cells E and F, thus, both of these two cells can be thought as the upwind cell, but both of these two cell centers are far from the normal line of the considered edge. Thus, the value of the upstream node is calculated by interpolating or extrapolating the values at the center of E and the center of F. There are two ways to draw a line to determine the position of the upstream node U: (1) the line is drawn through the edge center M in the normal direction of the edge (cf. Fig. 8a), so called UEM; or (2) the line is drawn through the cell center of C and the cell center of D (cf. Fig. 8b), so called UEQ. Note that in the latter case values are not reconstructed at the edge center M, but at the intersection point Q (cf. Fig. 8b). U is determined by intersecting this constructed line and a line drawn through the cell centers of E and F. It is noted that for UEM, the distance from cell center C to the projection point C_r ($|\overrightarrow{CC_r}|$) and the distance from cell center D to the projection point D_r ($|\overrightarrow{DD_r}|$) may be large in comparison to the distances $|\overrightarrow{C_rD_r}|$ and $|\overrightarrow{C_rU}|$, leading to numerical diffusion and increased mesh-dependency.

4.2. Improved vector manipulation method

In section 3.3, the discussion of occurring local extrema is based on Hou's vector manipulation method [13]. The MUSCL reconstruction interpolates the cell centered values to the edge along the gradient of the edge direction, or, in other words, along the outside neighbor cells. Special attention

has to be given to the downwind slope calculation in order to preserve the TVD property. In this study, it is suggested to use the geometric vector relationship to calculate the downwind slope by connecting both cells across the considered edge, and construct the vector from the outside cell center to this edge center, as shown in Fig. 9a. Then

$$\overrightarrow{CM} = \overrightarrow{CD} + \overrightarrow{DM} \tag{48}$$

so that,

$$\overrightarrow{r_{CM}} = \frac{|\overrightarrow{CD}|}{|\overrightarrow{CM}|}\overrightarrow{r_{CD}} + \frac{|\overrightarrow{DM}|}{|\overrightarrow{CM}|}\overrightarrow{r_{DM}} \tag{49}$$

$$\bigtriangledown q_{CM} = \frac{|\overrightarrow{CD}|}{|\overrightarrow{CM}|} \bigtriangledown q_{CD} + \frac{|\overrightarrow{DM}|}{|\overrightarrow{CM}|} \bigtriangledown q_{DM}. \tag{50}$$

The additional computational effort is very small compared to the scheme of Hou *et. al* [13], but the slope for calculating the downwind is solely depending on the variables of the downwind direction (outside cell). Recalculating the water depth in section 3.3, if $h_D = 1.0$, the downwind slope is calculated as $\bigtriangledown q_{CM} < 0.0$, and hence q_{CM} will be reconstructed first order accurate. If $h_D = 1.1$, the downwind slope is calculated by Eq. 49, giving $\bigtriangledown q_{CM} = -0.8064$. Then, q_{CM} will also be reconstructed first order accurate. When we observe the meshes, it can be seen that the line CM will pass through the center of cells D and E, and the cell E shows a decreased slope from cell C and D. Therefore, it is meaningful to use a first order for reconstruction of the values at the considered edge.

In comparison, consider the method by Buffard and Clain [8], where the slope is calculated along the line connecting the center of the considered cell with the center of the neighbor cell, cf. Fig. 9b. While this method is able to accurately calculate downwind slopes, the extrapolated values at the considered edge are calculated not in the center of this edge M, but rather at the intersection point Q and therefore may not represent the averaged values at the considered edge. Hereinafter, the improved vector manipulation method is referred to as VEM, and the method by Buffard and Clain is referred to as VEQ.

4.3. Comparison of the multislope MUSCL reconstruction procedures in a cell

The procedures of MUSCL reconstruction methods used in this work are summarized and compared in Tab. I.

In this work, a modified Van Leer's limiter function from [6] is adopted for 2D nodal evaluation methods:

$$\Psi(r) = \frac{0.5Rr + 0.5R|r|}{R - 1 + r} \tag{51}$$

Here, $R = (d_{Cf} + d_{Df})/d_{Cf}$; and a limiter function from [31] is adopted here for vector manipulation methods:

$$\Psi(a,b) = \begin{cases} \frac{(a^2+e)b+(b^2+e)a}{a^2+b^2+2e} & \text{if } ab > 0 \\ 0 & \text{if } ab \leq 0 \end{cases} \tag{52}$$

4.4. Boundary treatment

In this work, the boundary conditions are treated by following the description in [16], in order to maintain the high order inside the computational domain, a ghost cell technique is applied here. We will individually discuss the different MUSCL methods presented in Table I based on the legend of Fig. 10.

- UEQ: The ghost cells are used here for storing the values of the boundary information. In order to make sure that all internal cells have neighboring cells out of the local internal edge, the geometric information is stored by mirroring the internal neighboring cell. After storing the geometric information in the ghost cells, a two step boundary interpolation will be carried out to preserve the high order of accuracy for the whole computational domain.
  - The internal neighboring cells ACD, BDE will use cell centered values for calculating the ghost cells values by using the boundary conditions from [16].
  - The interpolation of edge values at edge CD and DE will be calculated based on the UEQ method, and the ghost cells' value will be chosen as the downwind cell value. After that, a new loop calculation for boundary edge values at DC and ED will be updated by the boundary conditions [16]. The ghost cells' values will remain the same as the boundary edges, and thus all internal cells and edges have neighboring cells for the high order reconstruction.
- VEM, VEQ and HOU: With the ghost cell technique, the slope from the center of ghost cell to the edge center will be thought as 0.0.
  - Compute the ghost cell values using the same method as UEQ. The downwind slope of the internal cell will be thought as the value difference between ghost cell and the internal cell divided by the distance from the internal cell center to the edge center and edge perpendicular point for the VEM (same as HOU) and VEQ, respectively.

4.5. A novel approach for restraining the recnstruction instability

The instability of MUSCL reconstructions is mainly caused by overestimated velocities [32], which will yield an overestimated flux across the cell edges. In order to avoid numerical instability, the velocities at the edge should satisfy the monotonicity condition

$$\max(u_C, u_D) \geq u_M^L \geq \min(u_C, u_D). \tag{53}$$

here, u_C, u_D and u_M^L means the velocities at the cell inside, outside and the velocity at edge after interpolation.

If the case that creates local maxima in velocity is considered, if the unit discharges have the same sign, i.e. $q_C q_D \geq 0$, then $\min(|q_C|, |q_D|) \leq |q_M^L| \leq \max(|q_C|, |q_D|)$, and Eq. 53 can be rewritten as

$$\min(|u_C|, |u_D|) \leq |u_M^L| \leq \max(|u_C|, |u_D|) \quad \text{if} \quad q_C q_D \geq 0. \tag{54}$$

This means, if the discharges have the same sign, the absolute value of the discharge can be considered for simplification. If the slope of the absolute value of the discharge and the slope of the water depth have different signs, no local maximum will be created, and therefore

$$\min(|u_C|,|u_D|) \leq |u_M^L| \leq \max(|u_C|,|u_D|) \quad \text{if} \quad \frac{d(|q|)}{dx} * \frac{dh}{dx} \leq 0 \wedge q_C q_D \geq 0. \tag{55}$$

It is always true for Eq. 55, besides this, if discharge with same signs are considered, two distinct cases can be identified that will create local extrema in the velocity: (1) increasing water depth, (2) decreasing water depth.

For the increasing water depth, if the velocity is beyond the range of the adjacent velocities, again two distinct cases have to be considered:

1. For the larger velocity interpolation: Velocities are calculated by using the discharge divided by the water depth. If the velocity is bigger than the adjacent velocities (absolute values), we can conclude that the discharge is overestimated. In this case, we will use the water depth h_M^L multiplied with the velocity with the larger absolute value. This means, that the increasing estimated situation (slope) of discharge is not changed, but adjusted by imposing a limit to the MUSCL reconstruction of the discharge, as shown in Fig. 11 (a).
2. For the smaller velocity interpolation: As before, we want to limit the MUSCL reconstruction values but do not want to change the slope sign for the MUSCL reconstruction. In the other word, the aim of the additional limitation is just for limiting the relatively bigger overestimate, for this smaller velocity interpolation, we think about the modification of water depth, increasing the absolute value of discharge may give local extrema, as shown in Fig. 11 (b).

For the decreasing water depth, if the velocity beyond the range of the velocities, we also give two situations for consideration.

- For the larger velocity interpolation: If the velocity is larger than the range of velocities (absolute values), the water depth is underestimated. The water depth will be calculated as the edge discharge q_M^L divided by the larger velocity of the both sides, as shown in Fig. 11 (c).
- For the smaller velocity interpolation: If the velocity is underestimated for the conditions of the water depth and the absolute discharge is decreasing, the discharge is underestimated so that the discharge will be calculated as the water depth multiplied with the smaller velocity of the both sides, as shown in Fig. 11 (d).

We can summarize as:

$$\text{if } q_C * q_D \geq 0 \begin{cases} q_M^L = h_M^L \max(|u_C|,|u_D|) q_M^L/|q_M^L|, & \text{if } dh/dx \geq 0, |u_M^L| > \max(|u_C|,|u_D|) \\ h_M^L = q_M^L/\min(|u_C|,|u_D|) q_M^L/|q_M^L|, & \text{if } dh/dx \geq 0, |u_M^L| < \min(|u_C|,|u_D|) \\ h_M^L = q_M^L/\max(|u_C|,|u_D|) q_M^L/|q_M^L|, & \text{if } dh/dx \leq 0, |u_M^L| > \max(|u_C|,|u_D|) \\ q_M^L = h_M^L \min(|u_C|,|u_D|) q_M^L/|q_M^L|, & \text{if } dh/dx \leq 0, |u_M^L| < \min(|u_C|,|u_D|) \end{cases} \tag{56}$$

The case for the both sides of the considered edge with different signs for the discharge is limited in a similar way. We analyzed the situation with the increasing and decreasing water depth, and as shown in Fig. 12 (a-d), all kinds of situations are listed which may occur for both sides with different signs for the discharges: the left side is the condition which will not create an extreme velocity and

the right side is the one where this situation may happen. The limitation can be summarized as:

$$\text{if } q_C * q_D < 0 \begin{cases} q_M^L = h_M^L \text{max}(u_C, u_D), & \text{if } dh/dx \geq 0, u_M^L > \text{max}(u_C, u_D) \\ q_M^L = h_M^L \text{min}(u_C, u_D), & \text{if } dh/dx \geq 0, u_M^L < \text{min}(u_C, u_D) \\ h_M^L = q_M^L / \text{max}(u_C, u_D), & \text{if } dh/dx \leq 0, u_M^L > \text{max}(u_C, u_D) \\ h_M^L = q_M^L / \text{min}(u_C, u_D), & \text{if } dh/dx \leq 0, u_M^L < \text{min}(u_C, u_D) \end{cases} \tag{57}$$

It can be observed that the water depth is modified during the limitation of the velocity. In two-dimensional shallow water equations, the unit discharge is usually splitted in x- and y-directions, which may make the limitation process more challenging, because the unit discharge in both directions can be limited independently. The modified water depth may lead to a local extremum in velocity in the direction that is not being limited, e.g. if the x-direction is limited a local extremum in the velocity in y-direction may occur. However, the treatment described in Eqs. 56 and 57 restricts the slope of the water depth while its sign is ensured to stay the same. Thus, the method for reconstructing the discharges in x- and y-direction will automatically select the smaller slope and therefore always satisfies the condition in both x- and y-direction. An additional challenge is that the variables have to be also reconstructed at the vertices of the cell, hence the limiting process is applied to the whole cell. Consequently, the order of accuracy of the MUSCL reconstruction decreases and if the values of discharge at the vertices differ significantly from the values at the cell edge, the interpolation of the discharge will be wrong. In order to control this issue, we propose to use a threshold value ϵ_+ for limiting very high velocities at the wet and dry interface. We choose this value to be $\epsilon_+ = 100\epsilon$, where $\epsilon = 10^{-6}$ m is the threshold that determines whether a cell is wet (if the water depth in the cell is larger than ϵ) or dry (if the water depth in the cell is less than ϵ), cf. e.g. [19, 21, 32].

5. COMPUTATIONAL EXAMPLES

We present 5 computational test cases, 2 with analytical solutions. The performance of the improved MUSCL reconstructions are compared regarding accuracy, efficiency and ease of implementation.

The first test case is the well-known Thacker's solution benchmark, which is chosen to demonstrate the accuracy and capability to deal with wet and dry interfaces of the MUSCL schemes. The second test case is chosen to verify the capability of the scheme to capture shock waves for high-speed discontinuous flow conditions and shows a steady-state oblique jump. The third test case shows a radial dam break and the fourth test case shows a 2D Riemann problem, to demonstrate the accuracy of the presented reconstruction methods. Finally, in a last test case a Tsunami experiment is replicated to demonstrate a near to real-world application.

Three types of meshes are employed in this work, the diagonal mesh, Scottish mesh and Delaunay mesh, respectively ([8]) as shown in Fig. 13.

5.1. Thacker's planar rotation benchmark

Thacker's planar rotation benchmark is considered to be one of the most challenging test cases for numerical codes, because it involves moving wet and dry fronts inside a parabola. The bottom topography is defined as

$$z_b(x,y) = -h_0\left[1 - \frac{(x-x_0)^2 + (y-y_0)^2}{a^2}\right] \tag{58}$$

where (x_0, y_0) represent the coordinates of the parabola center; h_0 is the water depth at the parabola center; a is the distance from the center point to the zero elevation of the shoreline. The analytical solution of this test case is given as

$$\eta_t = \frac{\sigma h_0}{a^2}\left[2(x-x_0)\text{cos}(\omega t) + 2(y-y_0)\text{sin}(\omega t)\right] \tag{59}$$

$$u(t) = -\omega\text{sin}(\omega t), \quad v(t) = \omega\sigma\text{cos}(\omega t) \tag{60}$$

where σ is a constant value, and $\omega = \sqrt{2gh_0}/a$ is the angular velocity of the rotation. In this work, parameters are set to be the same with [13] with $h_0 = 0.1$ m, $a = 1.0$ m, $\sigma = 0.5$ and the computational domain is set to be $4 \times 4\ m^2$ with the domain center at (2 m, 2 m). All boundaries are closed boundaries.

The characteristic length $\Delta x = \sqrt{A/N}$ is used to set the resolution of the meshes, A is the area of the computational domain and N is the total number of cells.

The accuracy is expressed as the L_1-error which can be calculated as

$$L_1 = \frac{\sum_1^n |q_i - q_{i,\text{exact}}|A_i}{\sum_1^n A_i}. \tag{61}$$

The upper part of Fig. 14 - 18 plot the water depth contour of FOU, HOU, UEQ, VEQ and VEM compared to the analytical solution, respectively, at $t = 3.5T$ and $4T$, T represents one period. It can be observed that VEM reaches the best agreement. HOU also yields good agreement, which is slightly less accurate than VEM. FOU yields the worst agreement and has more diffusion compared to other schemes; the lower part of the Fig. 14 - 18 show the velocity field plot, where it can be observed that the FOU gives the most drag effect of velocity, HOU shows a significantly drag tail, which maybe because of too much information from upwind, which leads to more diffusive behavior and VEM shows the best velocity field, where the vectors of the velocity almost coincide with the water depth contour. In order to compare the accuracy of different methods, we consider the worst grids to test the schemes. Cut section plots are shown in Fig. 19 and 20 for Delaunay and Scottish grids presented in Fig. 13. It is seen that VEM can obtain good accuracy on both grids, the difference between VEM and other methods is not very large on Delaunay grid; but on the Scottish grid, VEM yields much better agreement than the other methods. HOU always gives the second best agreement. The time step size is set adaptively using the CFL criterion, therefore the velocity influences the time step size. Even though the algorithm of VEM is slightly more complex than HOU, the computational effort is about 10% less than HOU, because no high velocities are reconstructed.

We present a mesh convergence study for this test case in Fig. 21 - 23. The L_1-error at $t = 4T$ for h and q_y are plotted in these figures for different grids, in which the lines with symbols represent the

numerical results varying with the refining of the mesh level (represented by characteristic length Δx shown in Tab. II) and the solid line is the order of 2. It can be seen that the results for VEQ and HOU are the same for the diagonal grid, as the middle point and the intersection point will reach to the same coordinate on diagonal grid, this also can be found in [8]. For the numerical results on diagonal grid, all the numerical results give a relatively low order, however, the VEM still can obtain highest accuracy and numerical order. For the Scottish and Delaunay grid, HOU scheme increases the accuracy significantly, which means that the scheme is more or less mesh dependent. And VEM scheme keeps the highest accuracy and numerical order. UEQ and VEQ scheme give almost slightly different results for different grids, the advantage of each scheme is related to the mesh type, but is quite small. The presented MUSCL schemes add a limitation presented in Sec. 4.5 at wet and dry fronts. This is to prevent the occurrence of negative water depths. Consequently, wet and dry fronts reduce the overall accuracy of these schemes, and this can be thought as the explanation of why the accuracy order cannot reach 2 in this test case.

In summary, it can be observed that VEM yields the lowest error and highest order in all figures, HOU yields a slightly lower error than the other methods for the water depth, but also yields the most diffusive velocity field except FOU. VEQ and UEQ are similar, because their algorithms are both based using a line connecting both cell centers. However, from the results we can conclude that the difference is negligibly small. The implementation of UEQ is more straight-forward. Present results are compared to the results reported in [16, 31]. The cell numbers in these test cases are similar to the present one, therefore a comparison is meaningful. The VEQ yields less accurate results, the VEM yields almost similar accuracy. It is noted that both the VEQ and VEM are computationally more efficient and more straightforward to implement than the approaches in [16, 31].

5.2. Steady oblique hydraulic jump

A steady-state hydraulic jump that develops when a unidirectional supercritical flow in an open channel hits an inclined solid wall is investigated. A $40\,\mathrm{m} \times 30\,\mathrm{m}$ frictionless domain with a flat bed is used and a uniform supercritical velocity with a Froude number of 2.7 is applied at the western inflow boundary. The initial water depth and velocity for the whole domain is set to $1\,\mathrm{m}$ and $8.57\,\mathrm{m/s}$, respectively. The eastern boundary is a free outflow boundary, and the northern and southern boundaries are closed boundaries. The southern wall is given an inclined angle of $8.95°$, beginning from $x = 10\,\mathrm{m}$ to the northeastern direction. The supercritical velocity will reflect, thus creating an oblique hydraulic jump along the southern wall. Theoretically, the water will jump from $1\,\mathrm{m}$ to $1.5\,\mathrm{m}$ and starting from $x = 10\,\mathrm{m}$, form a $30°$ to the x-direction. We enforce a constant time step for all methods for comparison purposes. The simulation runs for $10\,\mathrm{s}$ until the flow is steady and the hydraulic jump is formed. The simulation is carried out on a Delaunay grid (Fig. 13) with 3029 cells.

Results are shown in Fig. 24 - 28. The dashed line in the water level represents the $30°$ of the analytical solution. Except FOU all methods diminish the numerical diffusion. On the left of the figures, the cut sections show that the water depth increases from 1 m sharply to 10 m, and the water level almost reaches 1.4 m at $x = 5\,\mathrm{m}$. All MUSCL reconstruction methods yield good accuracy for this test case.

As shown in Tab. III, FOU is the fastest scheme. For the MUSCL reconstructions, UEQ is the most efficient method due to the straight-forward implementation and simple algorithm. Comparing to

HOU, VEQ only need to compute the linear system along the upwind direction, where can improve the computational efficiency. The VEM takes the most computational time, but compare to HOU, the difference is quite small, the additional treatment for including the downwind information doesn't influence the computational efficiency too much.

5.3. Radial dam break

A 2D radial dam break case from [33] is simulated to test the capability of the reconstruction methods to preserve the symmetry of the problem. The initial conditions are:

$$h(0,x,y) = \begin{cases} 2\ \mathrm{m} & \text{if } \sqrt{(x-25)^2+(y-25)^2} < 20.0 \\ 0.5\ \mathrm{m} & \text{otherwise} \end{cases} \tag{62}$$

The computational domain is a $50\,\mathrm{m} \times 50\,\mathrm{m}$ square. The computational mesh is a Delaunay mesh (Fig. 13) with 11932 cells. The simulation is run for $3\,\mathrm{s}$.

A reference solution is obtained using a high-resolution simulation with 1000000 cells on a structured grid using the high-resolution Godunov-type scheme of Clawpack [34].

Fig. 29 shows a 3D plot of VEM results at $t = 1.0$ s and $t = 2.5$ s. The cut section along $y = 25$ m is shown in Fig. 30, the numerical results are plotted over a section that goes from $(25, 50)$ m to $(50, 50)$ m at times $t = 1$ s, $t = 2$ s and $t = 3$ s.

The difference between the different MUSCL reconstructions are quite small, but at $t = 1$ s, it can be observed that the UEQ shows the best agreement, and VEM is slightly better than VEQ and HOU. The shock wave position is correctly captured, but due to numerical diffusion the shock wave front smeared. This is because of the low resolution of the grids. For $t = 2$s, the water level at the domain center is only well captured by HOU and VEM, while UEQ shows the lowest water level (about 0.05 m lower than the reference solution). Except at the center position, the MUSCL reconstructions are near to the reference solution, in which, the VEM shows the best agreement, followed by HOU. For $t = 3$ s, the shock wave is still well captured, but for the water level at domain center, HOU shows the most numerical diffusion and VEQ gives the best agreement with the maximum water level of the reference solution, followed by VEM.

5.4. 2D Riemann problem

This test case is originally presented in [4]. A frictionless $[0, 200] \times [0, 200]$ computational domain with a initial condition set as

$$h = \begin{cases} 10 & \text{if } x \geq 100 \wedge y \geq 100 \\ 1 & \text{else} \end{cases} \tag{63}$$

$$u = \begin{cases} 10 & \text{if } x \leq 100 \\ 0 & \text{else} \end{cases} \tag{64}$$

$$v = \begin{cases} 10 & \text{if } y \leq 100 \\ 0 & \text{else} \end{cases} \tag{65}$$

is used. The Delaunay computational mesh(Fig.13) consists of 6552 cells. In order to investigate the accuracy, a mesh that can be considered 'poor' is used.

Fig. 31 shows the flow pattern calculated by VEM at 1 s, 3 s, and 5 s. It can be observed that the shock wave positions are well captured. Due to the poor grid, the front of the shock wave is smeared. The rarefaction wave at the Northeast of the domain is well resolved by the numerical scheme. The water depth contour plot and the velocity field are shown in Fig. 32. As there is no analytical solution for this test case, quantitative analysis can not be further conducted, but it can be obviously found that the present MUSCL schemes produce less diffusion than FOU scheme.

In order to verify the order of accuracy of the methods, a cut section along [0,0] [200,200] is set inside the domain. The water depth along this cut section is shown in Fig. 33. We can observe that the MUSCL reconstruction methods are quite similar, but UEQ captured a slightly steeper rarefaction wave compared to the others. According to the description in [28] and the theoretical and numerical analysis in [2], the rarefaction waves are likely to be dampened by low-order schemes. Then, we can conclude that UEQ has the best performance in modeling shock problems on unstructured grids.

5.5. Tsunami wave impact on a conical island

We replicate the laboratory experiment from [35] using the presented numerical schemes and the MUSCL reconstructions. The experiment features wet and dry fronts, uneven topography and very shallow water depths, which are challenging for a numerical method.

The experiment domain is a [0.0, 25.92] m $\times$ [0.0, 27.6] m rectangle (Fig. 34). A Delaunay grid of 15692 cells is used for discretization. An ideal conical island with the center located at $[x_0, y_0]$ = [12.96, 13.8]m is defined as

$$z(x,y) = \max[\max(0.625, 0.9 - 1/(4\sqrt{(x-x_0)^2 + (y-y_0)^2})), 0.0]. \tag{66}$$

The initial still water level of the domain is 0.32 m, the island is partially submerged inside the water, a wave maker is set at the inflow boundary with a varying water level relative to the initial still water level, and the velocity is set as

$$\eta(t) = H\text{sech}^2[\sqrt{3H/4\eta_0^3}\sqrt{g(\eta_0 + H)}(t - T)] + \eta_0 \tag{67}$$

$$u(t) = \frac{C(\eta(t) - \eta_0)}{\eta(t)} \tag{68}$$

$$v(t) = 0 \tag{69}$$

where H is the amplitude of the wave and η_0 is the still water depth; T denotes the time for the wave crest reach the domain. The wave working condition is chosen as $\eta_0 = 0.32$ m, $T = 2.45$ s, and $H = 0.064$ m. Friction is not take into account in this case. The simulation run time is 20 s.

Fig. 35 shows the maximum wave run up at the front, the side and the back of the island at $t = 9$ s, $t = 11$ s and $t = 13$ s. Fig. 36 shows oscillating solutions computed by VEM and HOU without the treatment for the velocity, respectively, which means that the velocity limitation for the wetting and drying front significantly influences the stability of the numerical scheme.

Fig. 37 shows the comparison of measurement data from the experiment with the numerical results from different reconstruction methods at gauges located approximately at the run up area.

The gauges number 6, 9, 16 and 22 are located at [9.36, 13.80] m, [10.36, 13.80] m , [12.96, 11.22] m and [15.56, 13.80] m respectively, as shown in Fig. 34. Numerical results capture the peak of the water level at gauge 6 and gauge 9, but for the gauge 6, the trough is overestimated by all methods. For gauge 16 and gauge 22, both peak and trough are slightly smaller compared to the measurement data. This is because of the 3D effects of the wave propagation. The position of the wave is well captured by all the MUSCL reconstructions, FOU shows the most diffusion. Furthermore, the computed maximum run-up on the island is compared with the measurement data in Fig. 38. A slightly overestimated run-up can be observed for the MUSCL schemes at the front face to the wave come direction, this maybe the case because of the mesh size in relative to the wetting and drying interface which leads to a numerical error, while the backwash direction is well captured except for FOU. It can be observed that the difference of the results between the different MUSCL schemes are quite small, but all are much better than FOU. All MUSCL reconstructions are capable to handle wet and dry fronts over uneven terrain.

6. CONCLUSIONS

This paper presents two novel MUSCL reconstruction methods on unstructured grids: UEQ and VEM. Based on the TVD approach, the search for the upwind information is ambiguous. Therefore, we developed improved ways to determine the point from which the upwind information can be obtained. We derived an additional TVD condition, which limits the edge values based on the variables of the cells at the considered edge and showed that existing MUSCL reconstruction methods do not satisfy this condition. In order to avoid spurious velocity oscillations at the edge, we proposed a treatment for limiting the velocities.

The derived reconstruction methods are tested in 5 test cases. We compared results with the methods from [8, 13]. In most cases, we demonstrate that the novel VEM method is superior to the existing methods, especially on Scottish meshes (Fig. 13). The presented numerical scheme is able to handle wet and dry fronts, where the advantage and necessity of the proposed velocity treatment is significant.

ACKNOWLEDGEMENT

The authors gratefully acknowledge the China Scholarship Council for the scholarship granted to J. Zhao.

REFERENCES

1. van Leer B. Towards the ultimate conservative difference scheme. V. A second-order sequel to Godunov's method. *Journal of Computational Physics* jul 1979; **32**(1):101–136, doi:10.1016/0021-9991(79)90145-1.
2. Toro EF. *Riemann Solvers and Numerical Methods for Fluid Dynamics*. 3 edn., Springer-Verlag: Berlin Heidelberg, 2009.
3. Godunov SK. A difference method for numerical calculation of discontinuous equations of hydrodynamics (in Russian). *Matematicheskii Sbornik* 1959; **47**:271–300.
4. Guinot V. *Godunov-type Schemes: An introduction for engineers*. 1 edn., Elsevier Science B.V.: Amsterdam, The Netherlands, 2003.

5. Harten A. High resolution schemes for hyperbolic conservation laws. *Journal of Computational Physics* mar 1983; **49**(3):357–393, doi:10.1016/0021-9991(83)90136-5.
6. Hou J, Simons F, Hinkelmann R. Improved total variation diminishing schemes for advection simulation on arbitrary grids. *International Journal for Numerical Methods in Fluids* sep 2012; **70**(3):359–382, doi:10.1002/fld.2700.
7. LeVeque RJ. *Finite Volume Methods for Hyperbolic Problems*. 1 edn., Cambridge University Press: New York, USA, 2002.
8. Buffard T, Clain S. Monoslope and multislope MUSCL methods for unstructured meshes. *Journal of Computational Physics* may 2010; **229**(10):3745–3776, doi:10.1016/j.jcp.2010.01.026.
9. Venkatakrishnan V. Convergence to Steady State Solutions of the Euler Equations on Unstructured Grids with Limiters. *Journal of Computational Physics* apr 1995; **118**(1):120–130, doi:10.1006/jcph.1995.1084.
10. Darwish M, Moukalled F. TVD schemes for unstructured grids. *International Journal of Heat and Mass Transfer* feb 2003; **46**(4):599–611, doi:10.1016/S0017-9310(02)00330-7.
11. Li X, Liao H. An improved r-factor algorithm for TVD schemes. *International Journal of Heat and Mass Transfer* 2008; **51**(3-4):610–617, doi:10.1016/j.ijheatmasstransfer.2007.04.051.
12. Hou J, Simons F, Hinkelmann R. A new TVD method for advection simulation on 2D unstructured grids. *International Journal for Numerical Methods in Fluids* apr 2013; **71**(10):1260–1281, doi:10.1002/fld.3709.
13. Hou J, Liang Q, Zhang H, Hinkelmann R. Multislope MUSCL method applied to solve shallow water equations. *Computers & Mathematics with Applications* oct 2014; **m**, doi:10.1016/j.camwa.2014.09.018.
14. Liang Q, Borthwick AG. Adaptive quadtree simulation of shallow flows with wetdry fronts over complex topography. *Computers & Fluids* feb 2009; **38**(2):221–234, doi:10.1016/j.compfluid.2008.02.008.
15. Liang Q, Marche F. Numerical resolution of well-balanced shallow water equations with complex source terms. *Advances in Water Resources* 2009; **32**(6):873–884, doi:10.1016/j.advwatres.2009.02.010.
16. Hou J, Simons F, Mahgoub M, Hinkelmann R. A robust well-balanced model on unstructured grids for shallow water flows with wetting and drying over complex topography. *Computer Methods in Applied Mechanics and Engineering* apr 2013; **257**:126–149, doi:10.1016/j.cma.2013.01.015.
17. Bussing TRa, Murman EM. Finite-volume method for the calculation of compressible chemically reacting flows 1988; **26**(9):1070–1078, doi:10.2514/3.10013.
18. Hou J, Liang Q, Simons F, Hinkelmann R. A 2D well-balanced shallow flow model for unstructured grids with novel slope source term treatment. *Advances in Water Resources* feb 2013; **52**:107–131, doi:10.1016/j.advwatres.2012.08.003.
19. Simons F, Busse T, Hou J, Özgen I, Hinkelmann R. A model for overland flow and associated processes within the Hydroinformatics Modelling System. *Journal of Hydroinformatics* 2014; :1–26doi:10.2166/hydro.2013.173.
20. Liang Q, Marche F. Numerical resolution of well-balanced shallow water equations with complex source terms. *Advances in Water Resources* jun 2009; **32**(6):873–884, doi:10.1016/j.advwatres.2009.02.010.
21. Liang Q. Flood Simulation Using a Well-Balanced Shallow Flow Model. *Journal of Hydraulic Engineering* sep 2010; **136**(9):669–675, doi:10.1061/(ASCE)HY.1943-7900.0000219.
22. Buffard T, Clain S. Monoslope and multislope MUSCL methods for unstructured meshes. *Journal of Computational Physics* 2010; **229**(10):3745–3776, doi:10.1016/j.jcp.2010.01.026.
23. Guinot V, Delenne C. MUSCL schemes for the shallow water sensitivity equations with passive scalar transport. *Computers and Fluids* 2012; **59**:11–30, doi:10.1016/j.compfluid.2012.02.001.
24. Toro EF, Spruce M, Speares W. Restoration of the contact surface in the HLL-Riemann solver. *Shock Waves* jul 1994; **4**(1):25–34, doi:10.1007/BF01414629.
25. Audusse E, Bouchut F, Bristeau MO, Klein R, Perthame B. A Fast and Stable Well-Balanced Scheme with Hydrostatic Reconstruction for Shallow Water Flows. *SIAM Journal on Scientific Computing* jan 2004; **25**(6):2050–2065, doi:10.1137/S1064827503431090.
26. Roe PL. *Some contributions to the modeling of discontinuous flows*. 1985.
27. Venkatakrishnan V, Barth T. Application of direct solvers to unstructured meshes for the Euler and Navier-Stokes equations using upwind schemes. *27th Aerospace Sciences Meeting*, 1, American Institute of Aeronautics and Astronautics: Reston, Virigina, 1989; 5, doi:10.2514/6.1989-364.
28. Hou J, Liang Q, Zhang H, Hinkelmann R. An efficient unstructured muscl scheme for solving the 2d shallow water equations. *Environmental Modelling & Software* 2015; **66**:131–152.
29. Park JS, Yoon SH, Kim C. Multi-dimensional limiting process for hyperbolic conservation laws on unstructured grids. *Journal of Computational Physics* 2010; **229**(3):788–812, doi:10.1016/j.jcp.2009.10.011.
30. Bruner C, Walters R. Parallelization of the Euler equations on unstructured grids. *13th Computational Fluid Dynamics Conference* 1997; :446–470doi:10.2514/6.1997-1894.
31. Delis A, Nikolos I. A novel multidimensional solution reconstruction and edge-based limiting procedure for unstructured cell-centered finite volumes with application to shallow water dynamics. *International Journal for*

Numerical Methods in Fluids 2013; **71**(5):584–633.

32. Hou J, Liang Q, Simons F, Hinkelmann R. A stable 2d unstructured shallow flow model for simulations of wetting and drying over rough terrains. *Computers & Fluids* 2013; **82**:132–147.
33. Canestrelli A, Dumbser M, Siviglia A, Toro EF. Well-balanced high-order centered schemes on unstructured meshes for shallow water equations with fixed and mobile bed. *Advances in Water Resources* mar 2010; **33**(3):291–303, doi:10.1016/j.advwatres.2009.12.006.
34. Clawpack Development Team. Clawpack software 2014, doi:10.5281/zenodo.50982. URL `http://www.clawpack.org`, version 5.3.1.
35. Briggs MJ, Synolakis CE, Harkins GS, Green DR. Laboratory experiments of tsunami runup on a circular island. *Tsunamis: 1992–1994*. Springer, 1995; 569–593.

Table I. Procedures of multislope MUSCL reconstructions

Steps	2D nodal evaluation Q method (UEQ). Choose Fig. 8b as legend	Vector manipulation extrapolate Q method (VEQ) Choose Fig. 9b as legend
1	Calculate the coordinate of intersection point U of line CD and the neighbor cell centers connecting line EF	Compute $\overrightarrow{r_k}$ and $\overrightarrow{k}$
2	Interpolate value U from values at center E and F	Solve Eq. (22) get the coefficient β_1 and β_2.
3	Extrapolate the edge value by following Eqs. (16, 17)	Calculate $\bigtriangledown q_{C,(D,F,H)}$ from Eq. (20)
4	-	Extrapolate $\bigtriangledown q_{N_3C}$ from Eq. (23)
5	-	Extrapolate the edge value by following Eq. (25)
Steps	Vector manipulation extrapolate M method (VEM). Choose Fig. 9a as legend.	Old vector manipulation extrapolate M method (HOU). Choose Fig. 9a as legend.
1	Compute $\overrightarrow{r_k}$ and $\overrightarrow{t_k}$	Compute $\overrightarrow{r_k}$ and $\overrightarrow{t_k}$
2	Solve Eqs. (21 & 22) to get the coefficient α_1 , α_2, β_1 and β_2.	Solve Eq. (21 & 22) to get the coefficient α_1 , α_2, β_1 and β_2.
3	Calculate $\bigtriangledown q_{C,(D,F,H)}$ and $\bigtriangledown q_{D,(C,E,G)}$ from Eq. (20)	Calculate $\bigtriangledown q_{C,(D,F,H)}$ and $\bigtriangledown q_{D,(C,E,G)}$ from Eq. (20)
4	Evaluate $\bigtriangledown q_{CM}$ and $\bigtriangledown q_{N_3C}$ from Eq. (23)	Evaluate $\bigtriangledown q_{CM}$ and $\bigtriangledown q_{N_3C}$ from Eq. (23)
5	Update $\bigtriangledown q_{CM}$ with Eq. (50)	Extrapolate the edge value by following Eq. (25)
6	Extrapolate the edge value by following Eq. (25)	-

Table II. Characteristic length Δx used for mesh convergence test

Mesh level	Diagonal mesh (m)	Scottish mesh (m)	Delaunay mesh (m)
1	0.0442	0.0435	0.0439
2	0.0314	0.0313	0.0315
3	0.0224	0.0222	0.0223
4	0.0159	0.0159	0.0156

Table III. Oblique hydraulic jump: relatively computational cost.

Schemes	FOU	UEQ	VEQ	HOU	VEM
Relative computational time	1.00	1.35	1.40	1.47	1.50

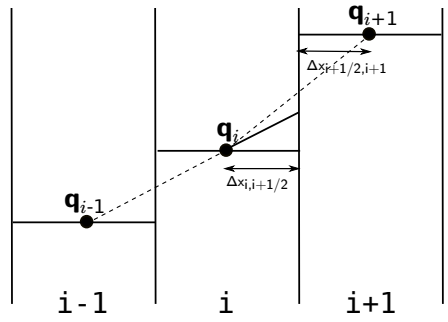


Figure 1. One dimensional MUSCL reconstruction

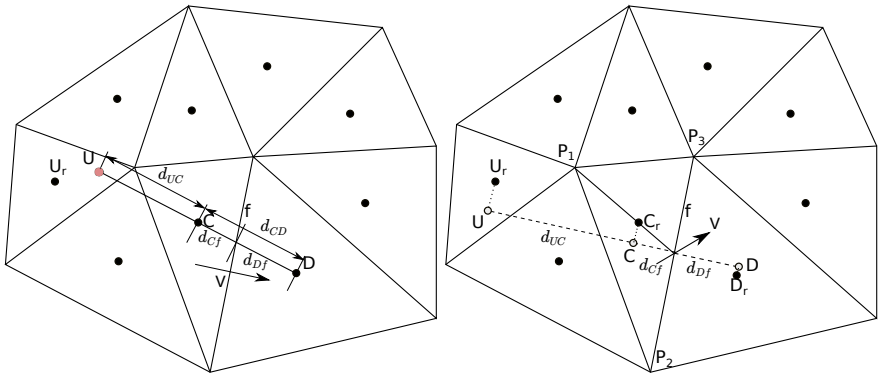


Figure 2. Three-point-stencil for Darwish and Moukalled [10] (U is not taken into account), Li and Liao [11] and Hou *et al.* [6] (left); three-point stencil for Hou *et al.* [12] (right)

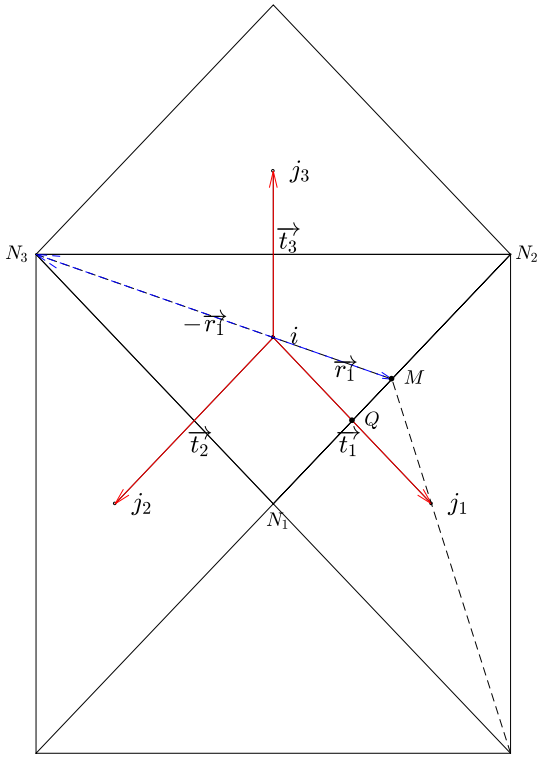


Figure 3. Vector notations at the considered and the neighboring cells

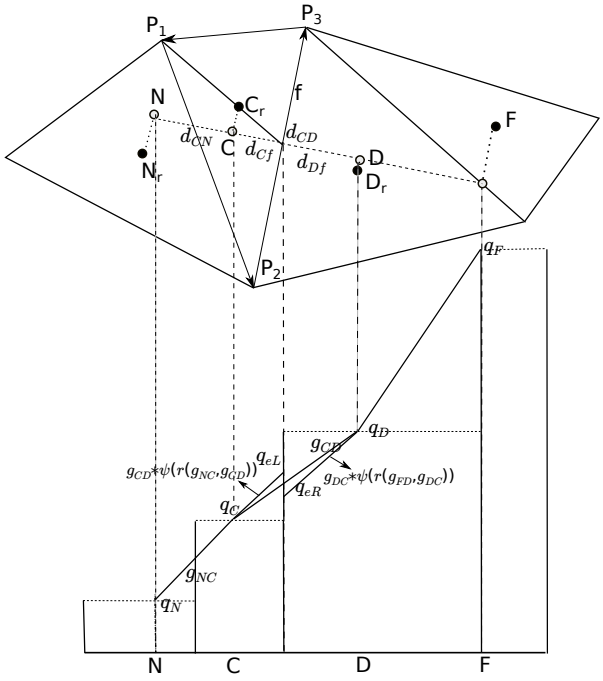


Figure 4. Monotonicity violating reconstruction

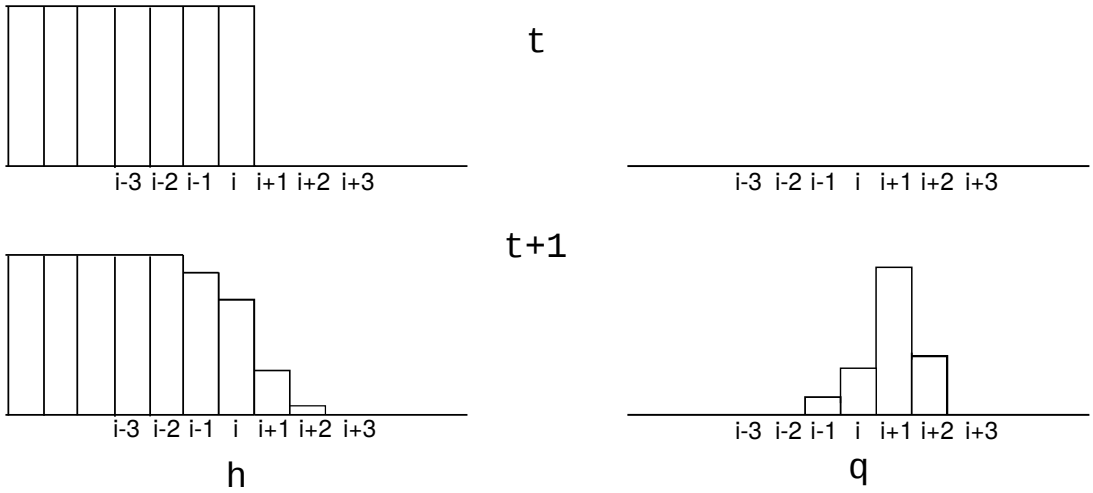


Figure 5. Illustration of interpolation error in 2D node evaluation methods

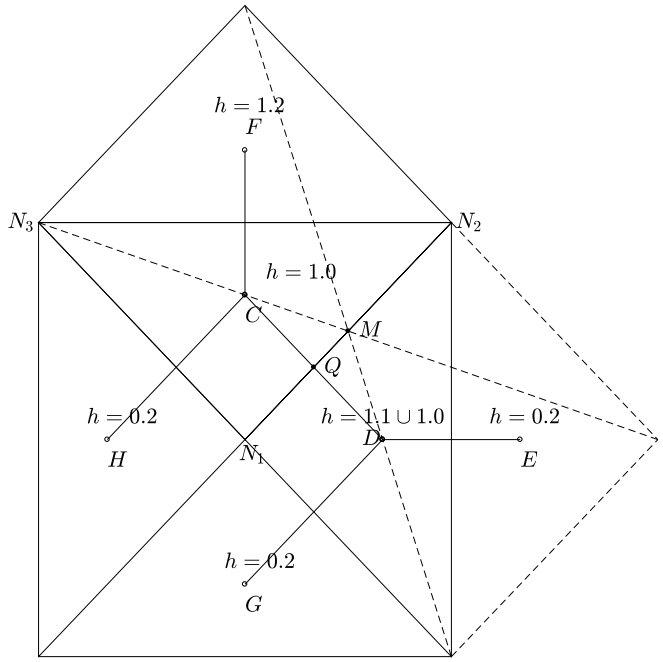


Figure 6. Stencils for vector manipulation method

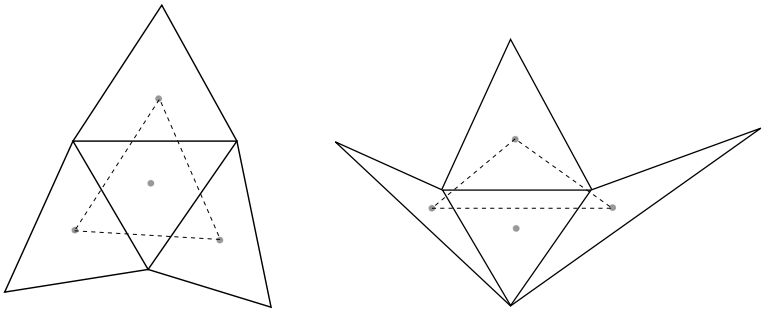
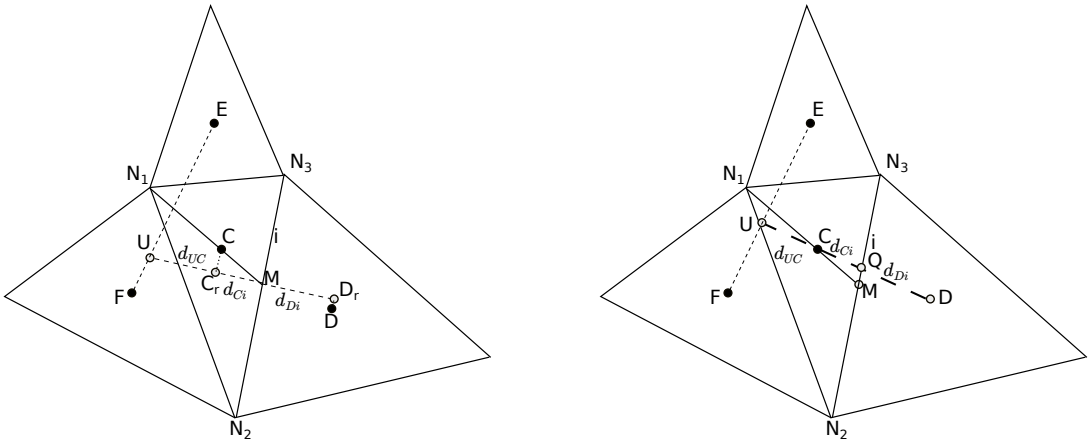
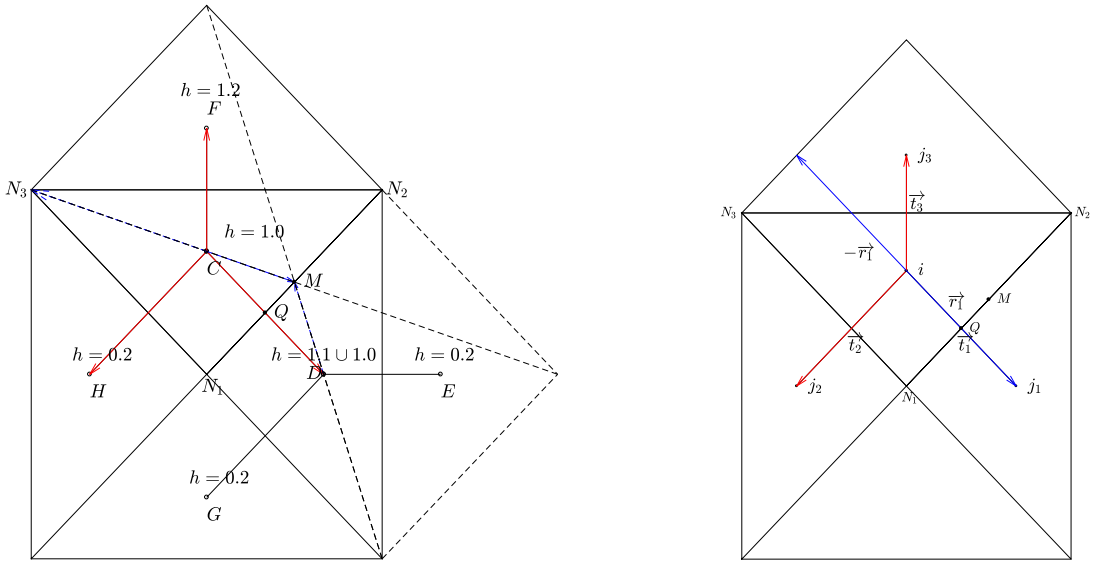


Figure 7. Configuration that satisfies (left) and does not satisfy (right) the assumption



(a) UEM (b) UEQ

Figure 8. Stencils for (a) 2D nodal evaluation method along edge normal direction (UEM); (b) 2D nodal evaluation method along the cell centerline (UEQ)



(a) VEM (b) VEQ

Figure 9. Stencils for (a) improved vector manipulation method (VEM); (b) Buffard and Clain's vector manipulation method (VEQ)

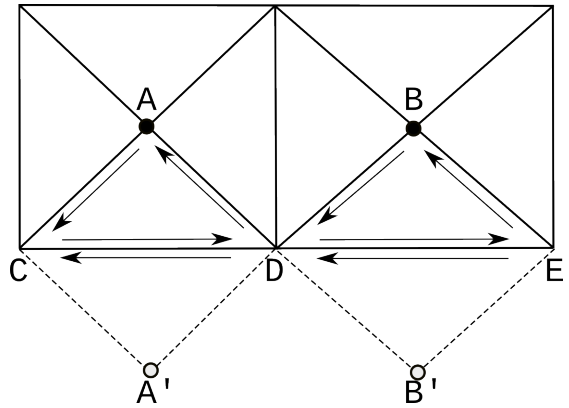


Figure 10. Ghost cell techniques used for the MUSCL reconstructions

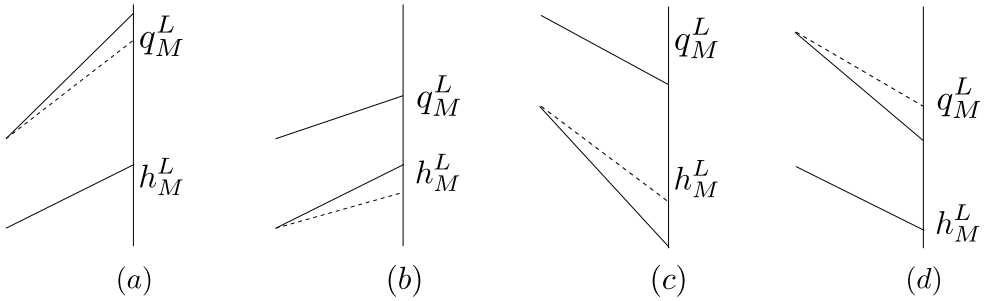


Figure 11. When $q_C q_D \geq 0$, the local extrema for velocities created by MUSCL reconstruction: (a) $dh/dx > 0$, $|u_M^L| > \max(|u_C|, |u_D|)$, the overestimated result is controlled by limiting the q_M^L; (b) $dh/dx > 0$, $|u_M^L| < \min(|u_C|, |u_D|)$, the overestimated result is controlled by limiting the h_M^L; (c) $dh/dx < 0$, $|u_M^L| > \max(|u_C|, |u_D|)$, the overestimated result is controlled by limiting the h_M^L; (d) $dh/dx < 0$, $|u_M^L| < \min(|u_C|, |u_D|)$, the overestimated result is controlled by limiting the q_M^L.

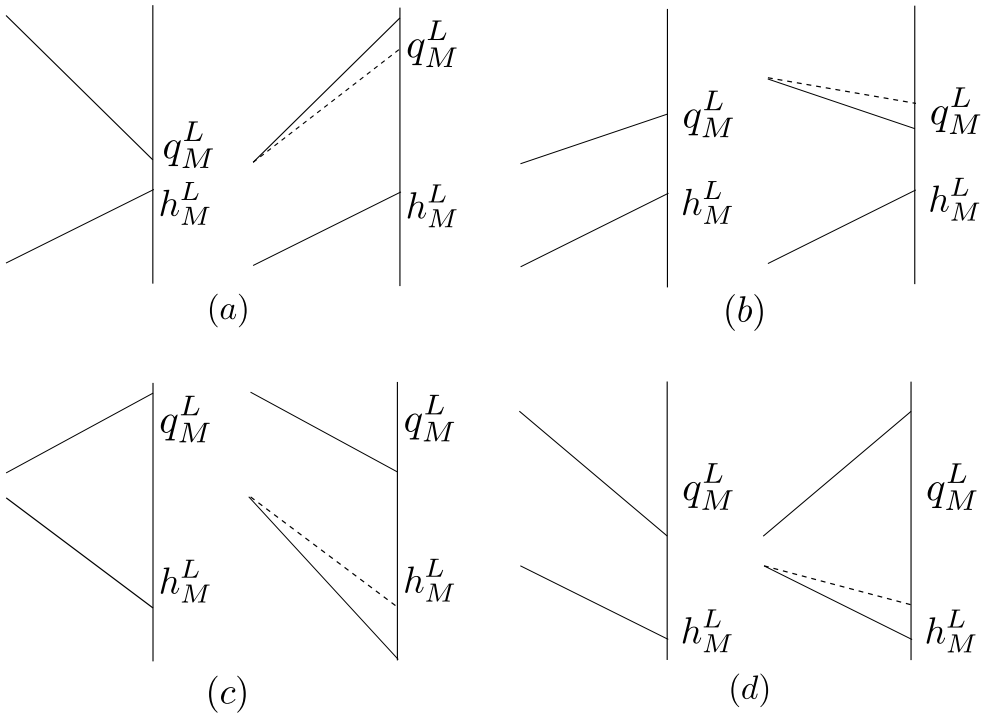


Figure 12. When $q_C q_D < 0$ all the left hand side of (a-d) shows the situations can not create the local extrema velocities, for the right sides: (a) $dh/dx > 0$, $u_M^L > \max(u_C, u_D)$, the overestimated result is controlled by limiting the q_M^L; (b) $dh/dx > 0$, $u_M^L < \min(u_C, u_D)$, the overestimated result is controlled by limiting the q_M^L; (c) $dh/dx < 0$, $u_M^L > \max(u_C, u_D)$, the overestimated result is controlled by limiting the h_M^L; (d) $dh/dx < 0$, $u_M^L < \min(u_C, u_D)$, the overestimated result is controlled by limiting the h_M^L.

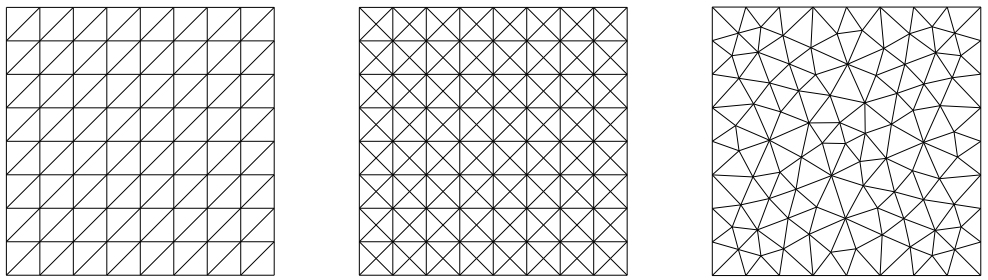
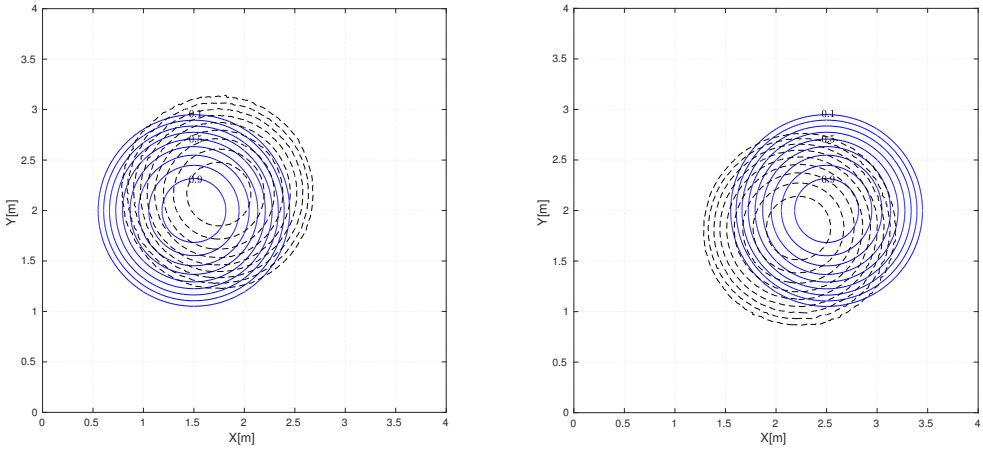
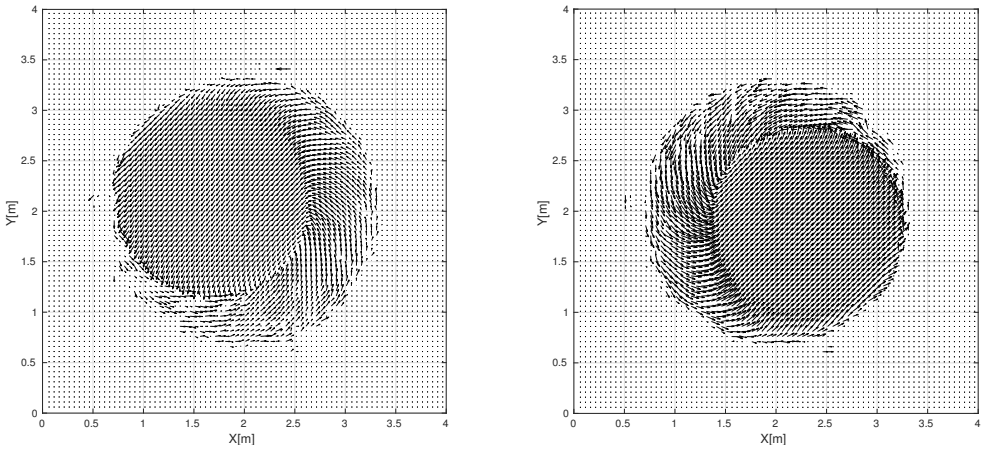


Figure 13. The three types of mesh employed to evaluate the effective accuracy of the schemes: the diagonal mesh (left), the Scottish mesh (center) and the Delaunay mesh (right)

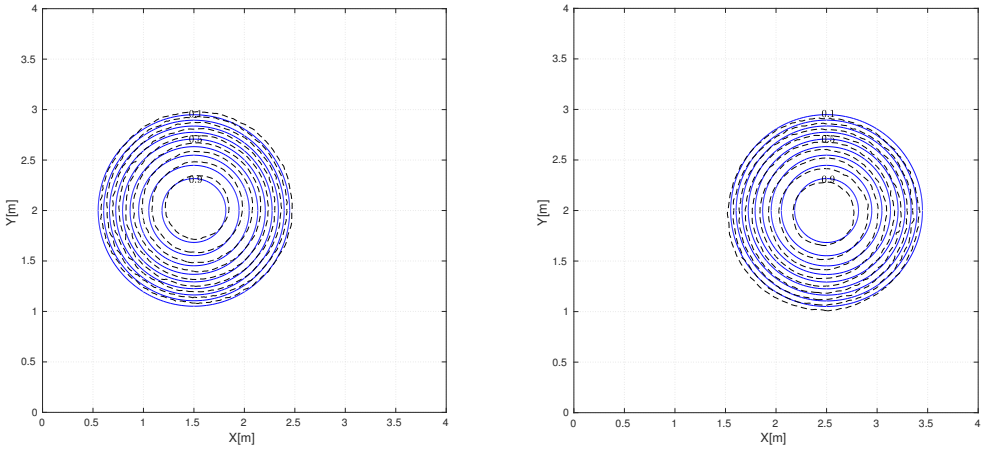


(a) Contour plot

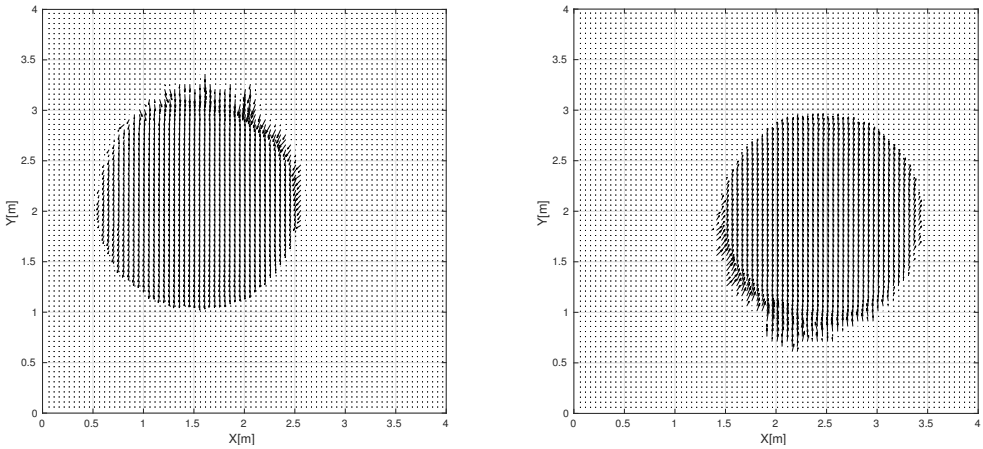


(b) Velocity field plot

Figure 14. Thacker's planar solution: contours of water depth computed by FOU scheme (dashed line) and the analytical solution (solid line) from Thacker at: t = 3.5T (a: left), t = 4.0 T (a: right), unit (m), with 16364 cells of Delaunay mesh. (b) The corresponding velocity field.

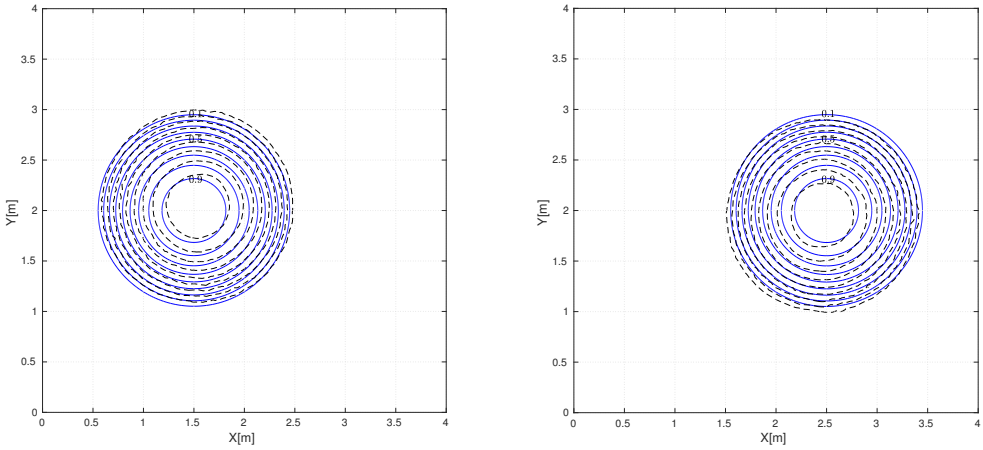


(a) Contour plot

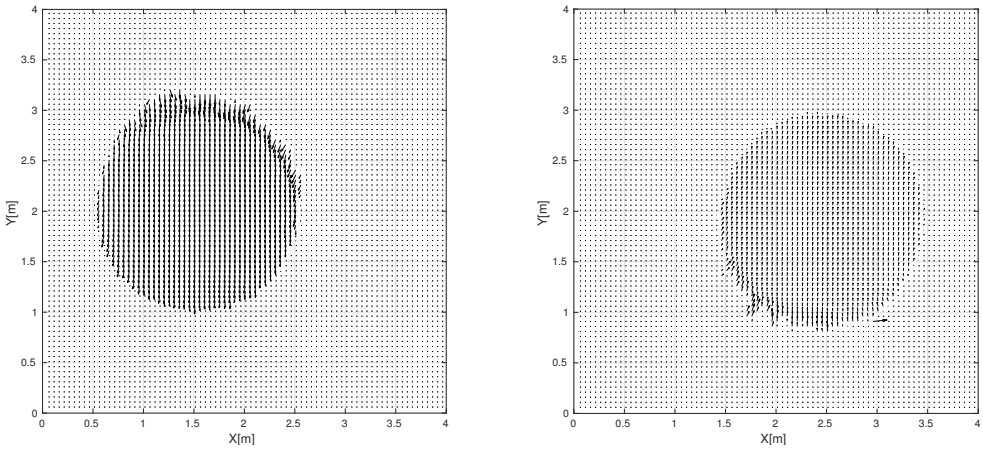


(b) Velocity field plot

Figure 15. Thacker's planar solution: contours of water depth computed by HOU scheme (dashed line) and the analytical solution (solid line) from Thacker at: t = 3.5T (a: left), t = 4.0 T (a: right), unit (m), with 16364 cells of Delaunay mesh. (b) The corresponding velocity field.

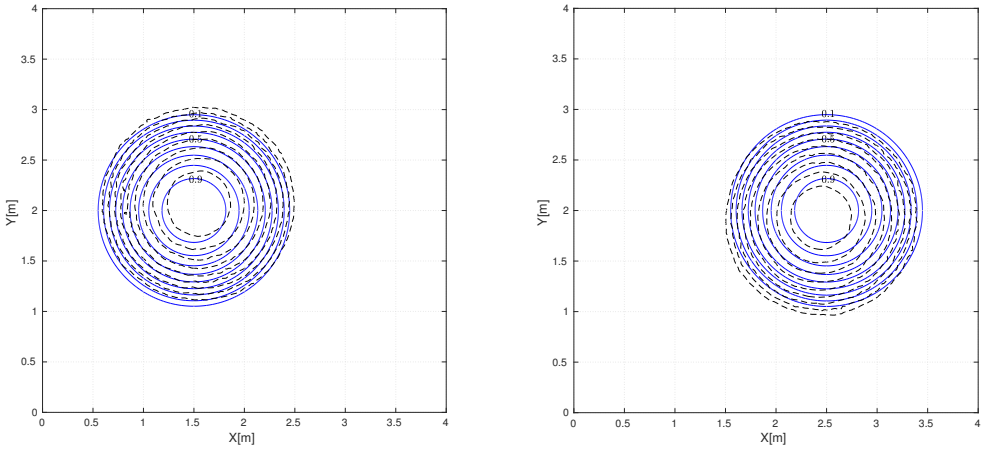


(a) Contour plot

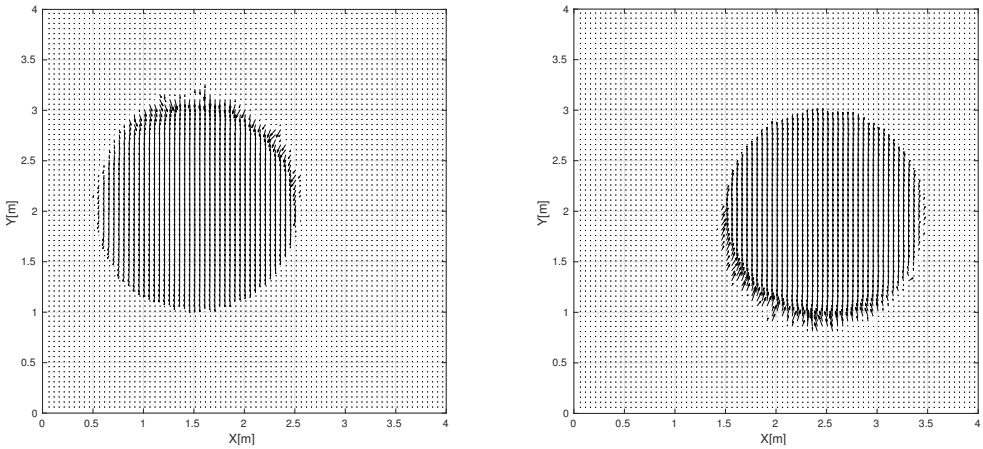


(b) Velocity field plot

Figure 16. Thacker's planar solution: contours of water depth computed by UEQ scheme (dashed line) and the analytical solution (solid line) from Thacker at: t = 3.5T (a: left), t = 4.0 T (a: right), unit (m), with 16364 cells of Delaunay mesh. (b) The corresponding velocity field.

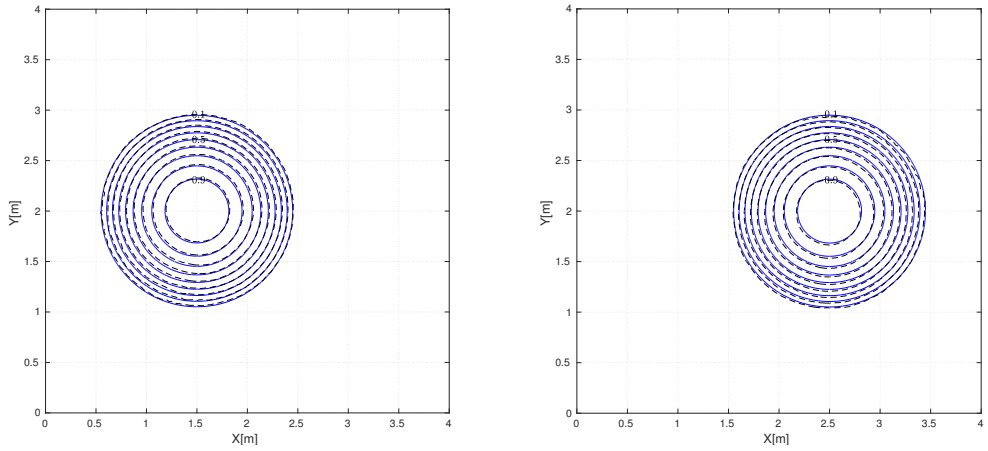


(a) Contour plot

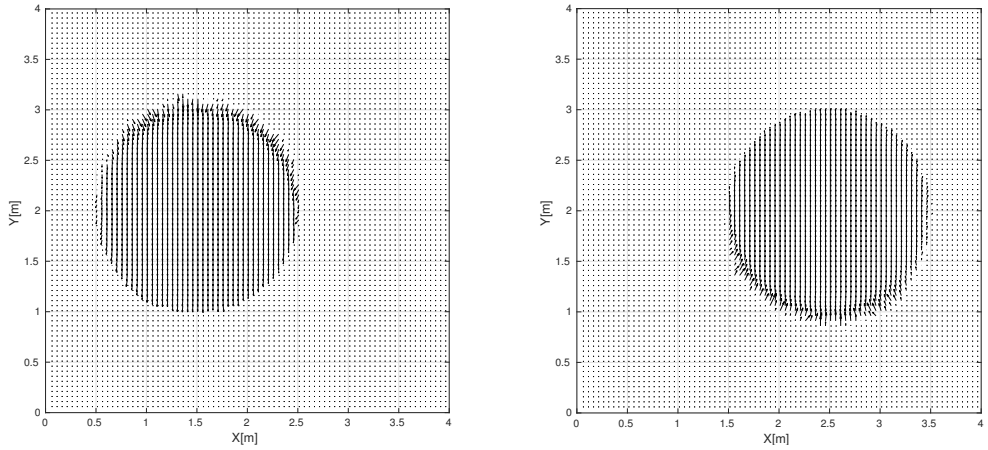


(b) Velocity field plot

Figure 17. Thacker's planar solution: contours of water depth computed by VEQ scheme (dashed line) and the analytical solution (solid line) from Thacker at: t = 3.5T (a: left), t = 4.0 T (a: right), unit (m), with 16364 cells of Delaunay mesh. (b) The corresponding velocity field.



(a) Contour plot



(b) Velocity field plot

Figure 18. Thacker's planar solution: contours of water depth computed by VEM scheme (dashed line) and the analytical solution (solid line) from Thacker at: t = 3.5T (a: left), t = 4.0 T (a: right), unit (m), with 16364 cells of Delaunay mesh. (b) The corresponding velocity field.

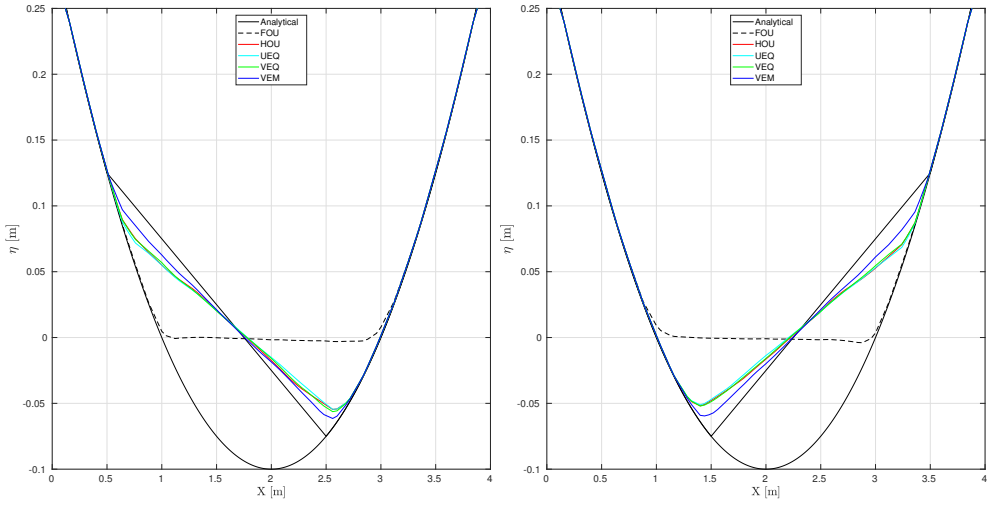


Figure 19. Thacker's planar solution: computed and analytical water levels at cross section of $y = 2$ m at: t = 3.5 T (left), t = 4.0 T (right), with 2062 Delaunay meshes.

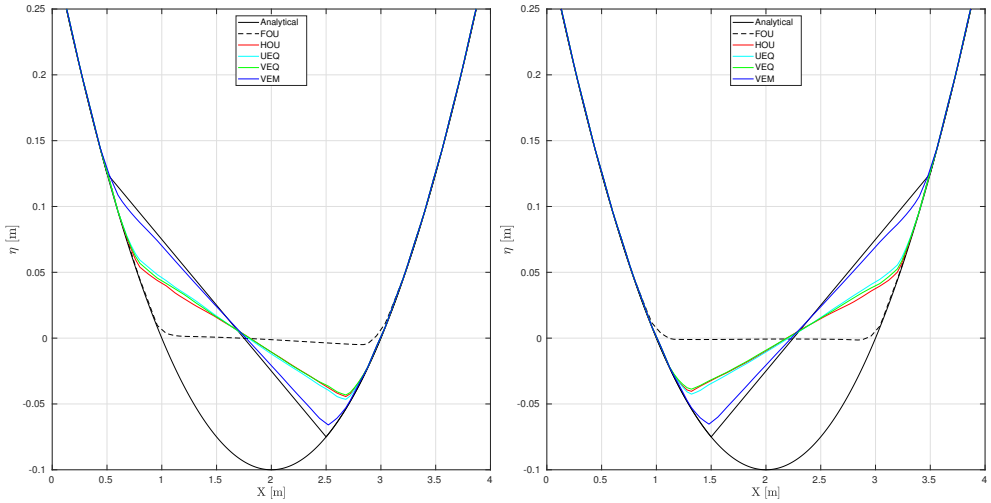


Figure 20. Thacker's planar solution: computed and analytical water levels at cross section of $y = 2$ m at: t = 3.5 T (left), t = 4.0 T (right), with 2116 Scottish meshes.

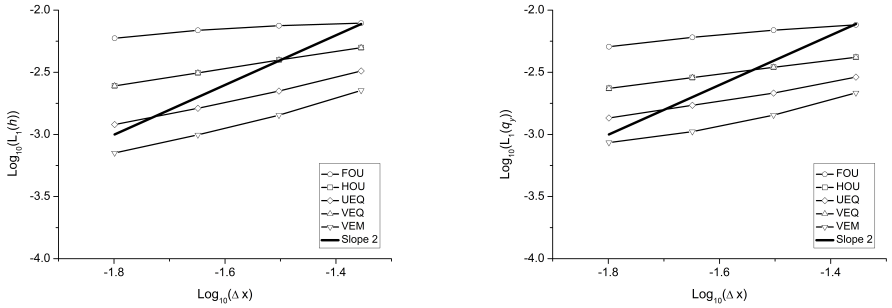


Figure 21. Thacker's planar solution: h (left) and q_y (right) convergence results for numerical schemes used on diagonal grid at t = 4 T.

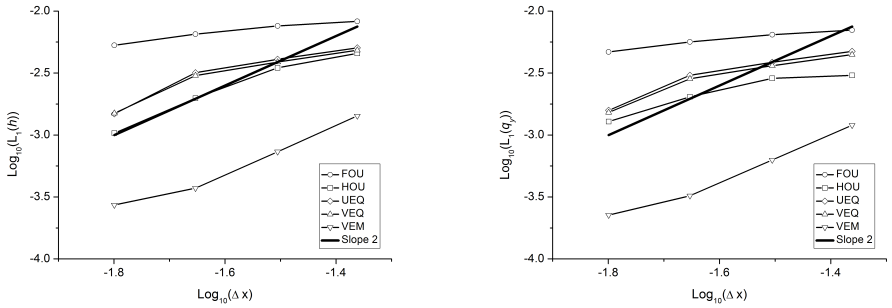


Figure 22. Thacker's planar solution: h (left) and q_y (right) convergence results for numerical schemes used on Scottish grid at t = 4 T.

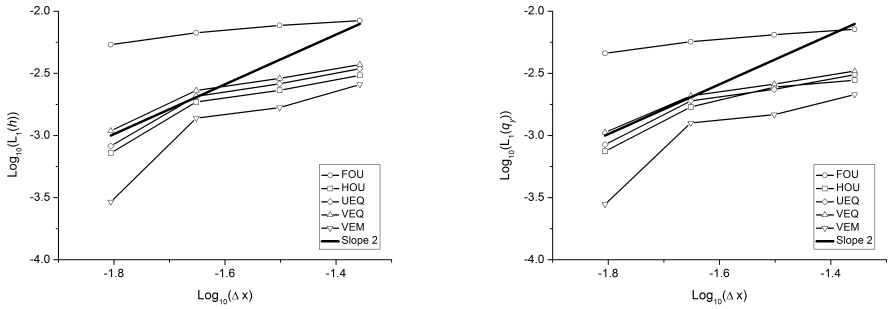


Figure 23. Thacker's planar solution: h (left) and q_y (right) convergence results for numerical schemes used on Delaunay grid at t = 4 T.

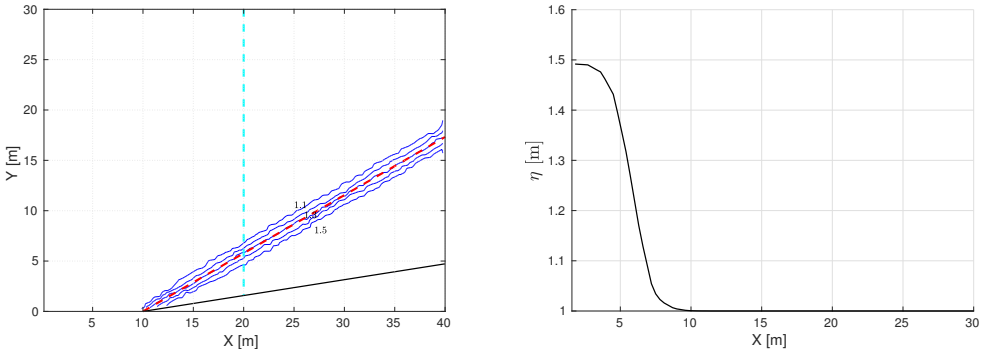


Figure 24. Water level contour (left) and cut section plot for FOU scheme (right).

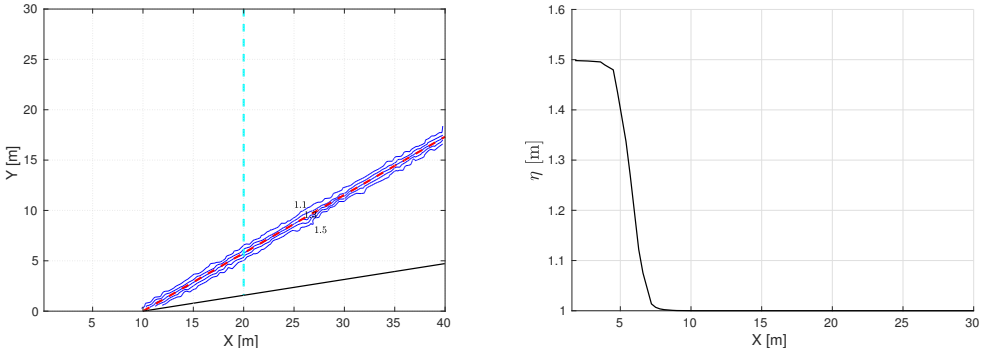


Figure 25. Water level contour (left) and cut section plot for HOU scheme (right).

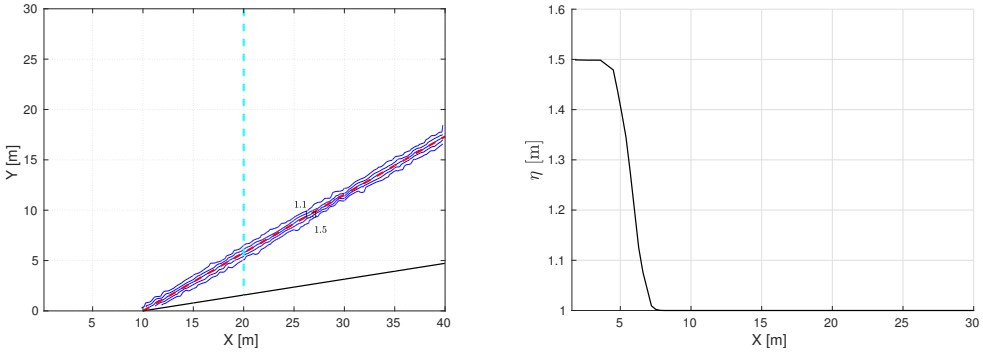


Figure 26. Water level contour (left) and cut section plot for UEQ scheme (right).

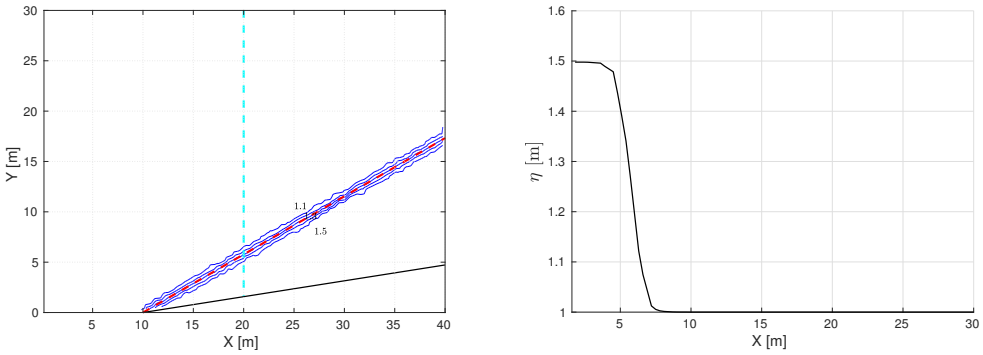


Figure 27. Water level contour (left) and cut section plot for VEQ scheme (right).

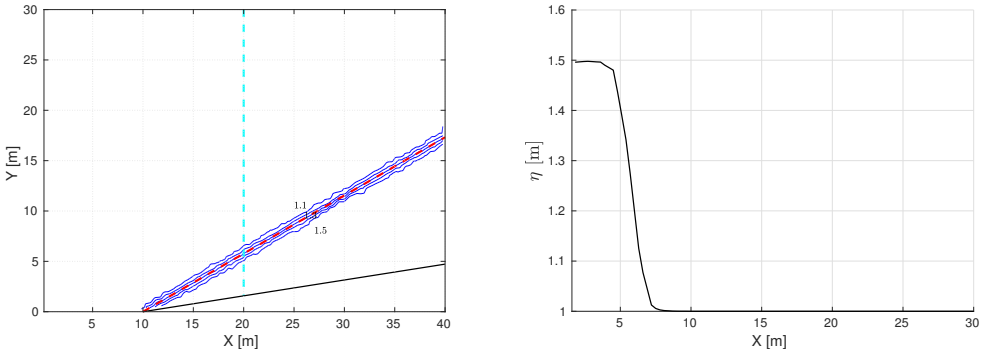


Figure 28. Water level contour (left) and cut section plot for VEM scheme (right).

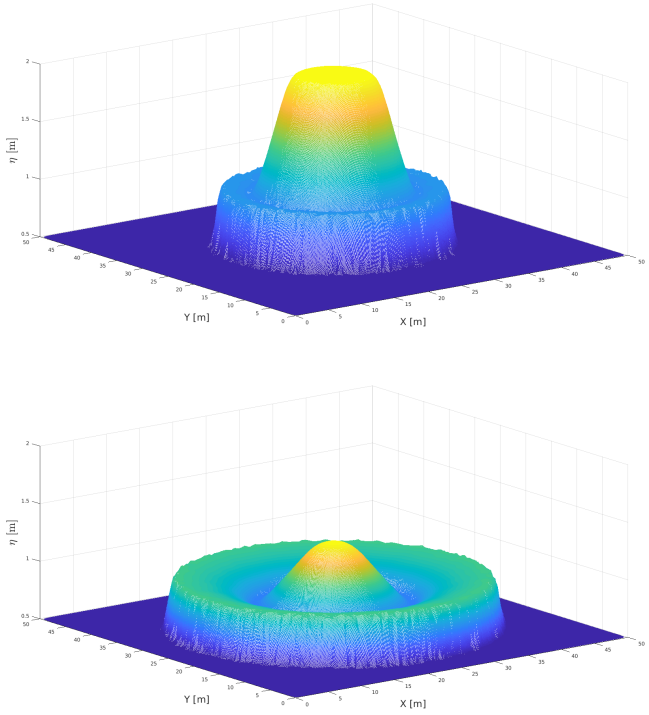


Figure 29. Radial dam-break problem: 3D view of water level computed by VEM at: t = 1.0 s (upper); t = 2.5 s (lower).

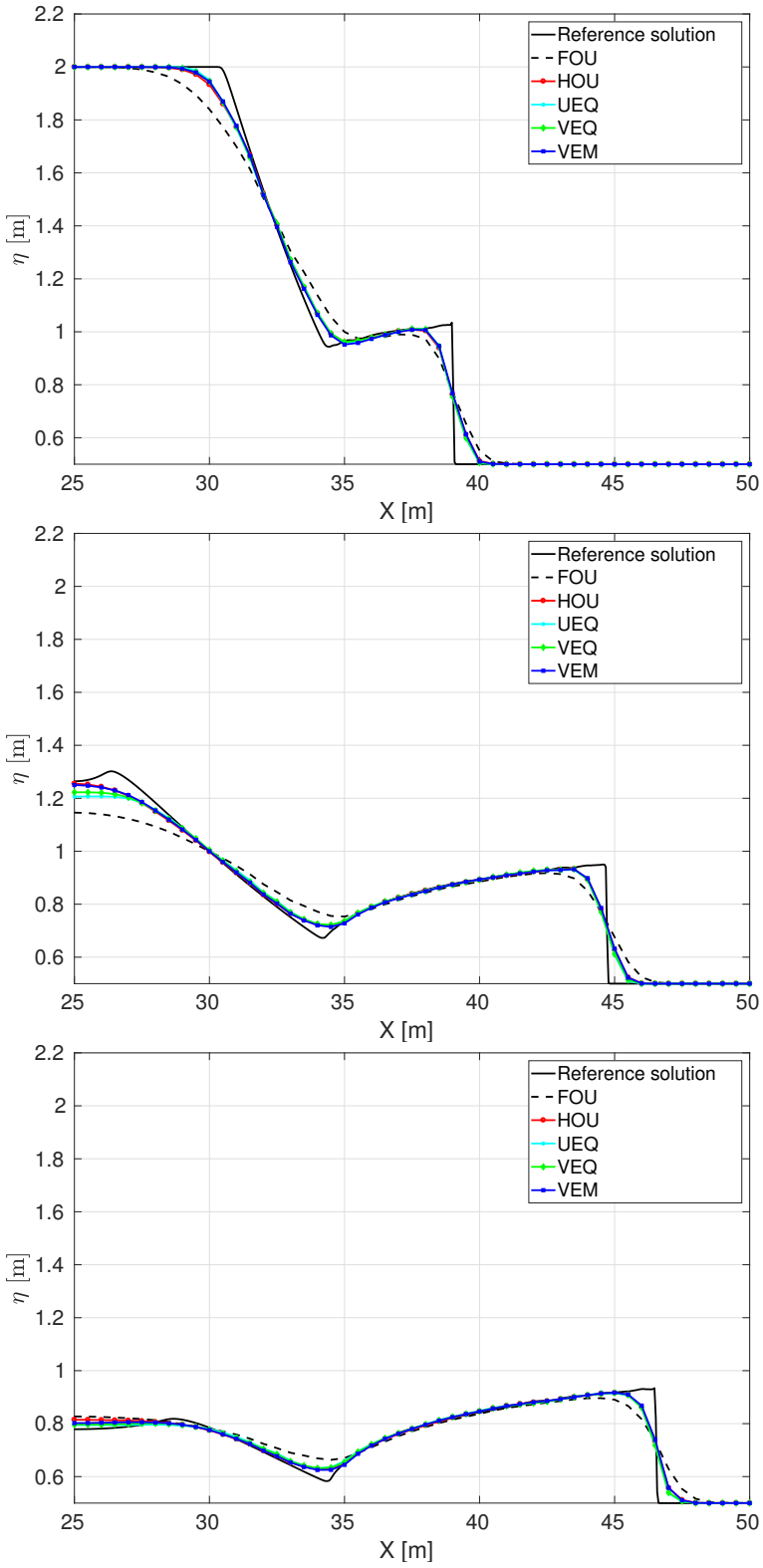
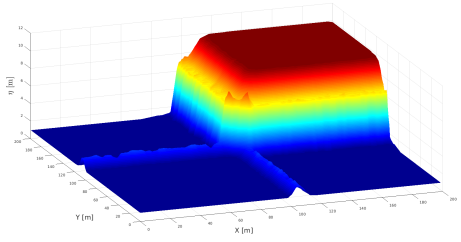
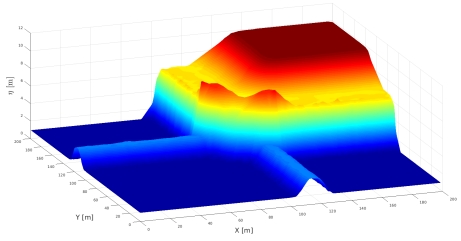


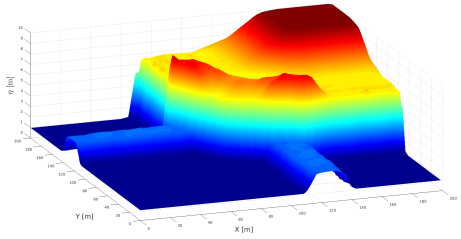
Figure 30. Radial dam-break problem: section view computed water level at: t = 1.0 s (upper); t = 2.5 s (middle); t = 3.0 s (lower).



(a) t = 1s



(b) t = 3s



(c) t = 5s

Figure 31. 2D Riemann problem: 3D views of computational flow pattern by VEM scheme at: t = 1.0 s (a); t = 3.0 s (b); t = 5.0 s (c).

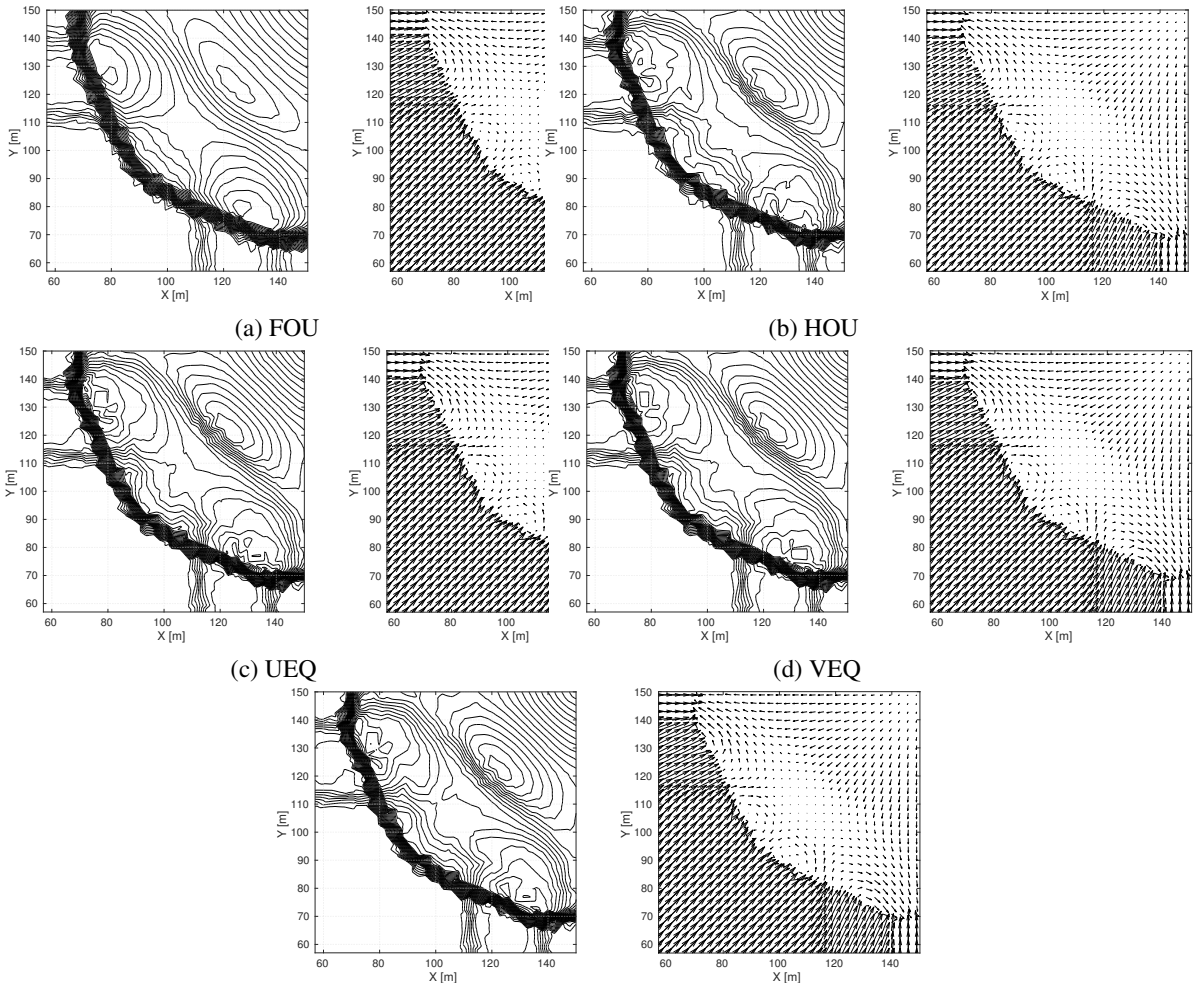


Figure 32. 2D Riemann problem: contour plot of water depths (left) and velocity vector fields (right) for different numerical schemes

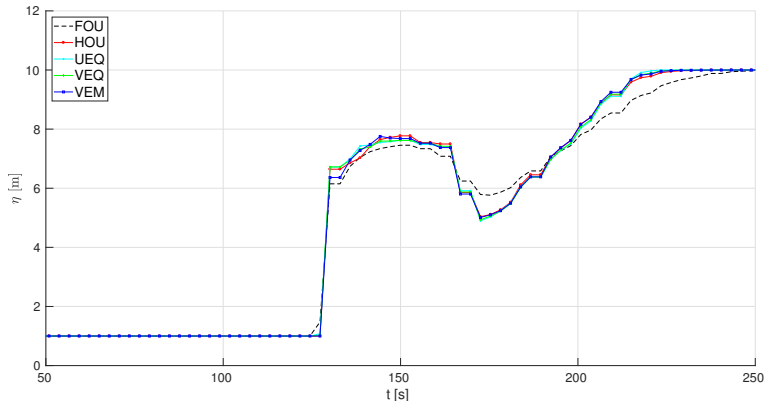


Figure 33. 2D Riemann problem: diagonal section water depth plot at t = 5s for different numerical schemes

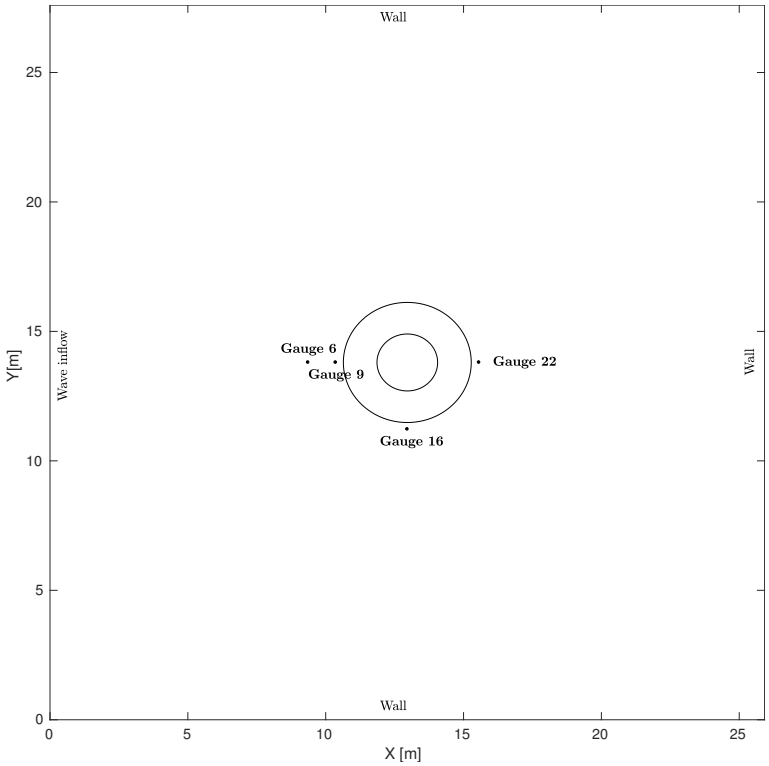
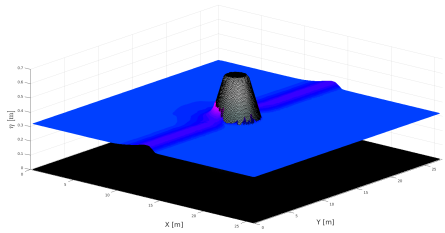
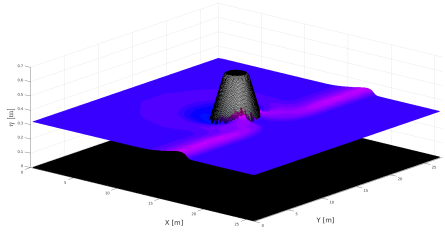


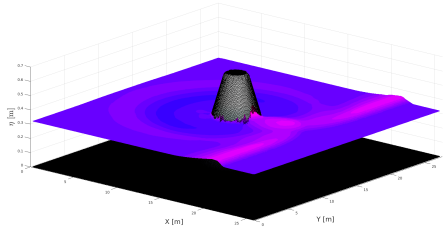
Figure 34. Tsunami wave impact on a conical island: computational domain, boundary conditions and locations of selected gauges.



(a) t = 9s

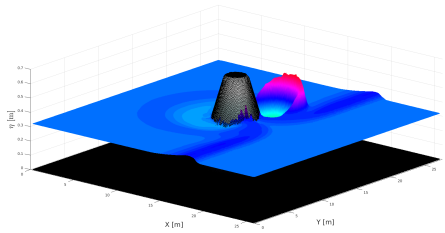


(b) t = 11s

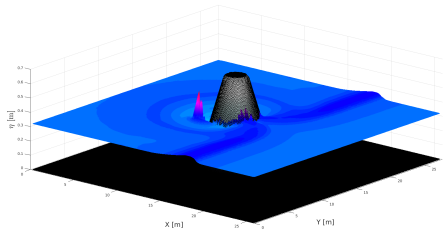


(c) t = 13s

Figure 35. Tsunami wave impact on a conical island: 3D view of water depth and the bottom topography calculated by UEQ scheme at: (a) t = 9s, (b) t = 11s and (c) t = 13 s.

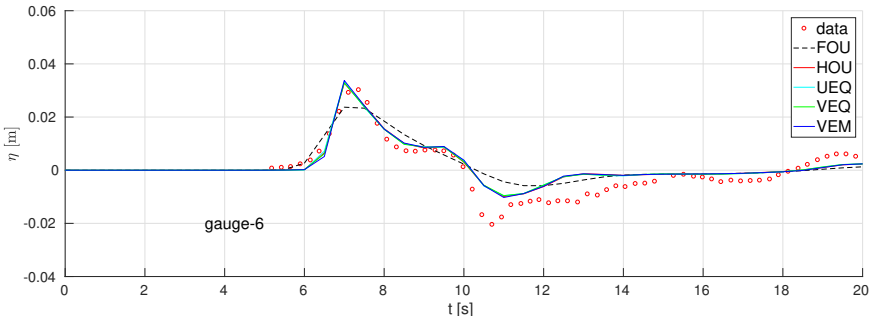


(a) VEM

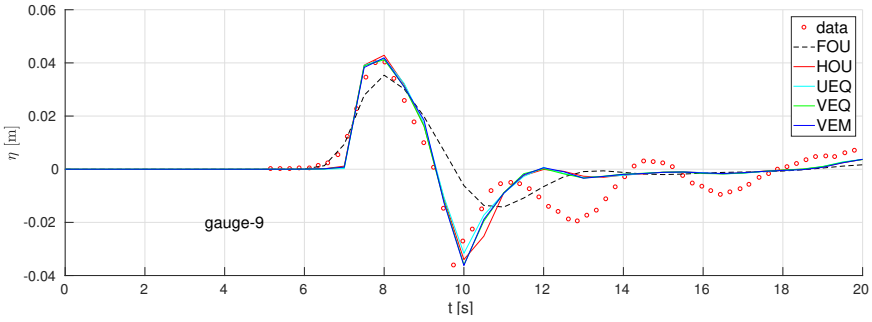


(b) HOU

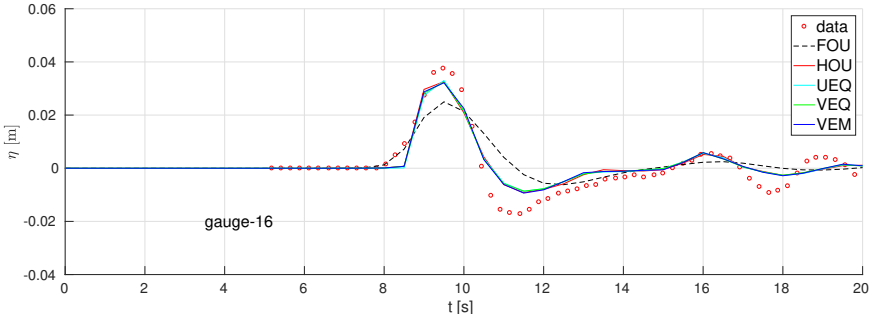
Figure 36. Tsunami wave impact on a conical island: 3D view of water depth and the bottom topography calculated by VEM and HOU scheme without limitation for the velocity at 11.5 s: (a) VEM and (b) HOU.



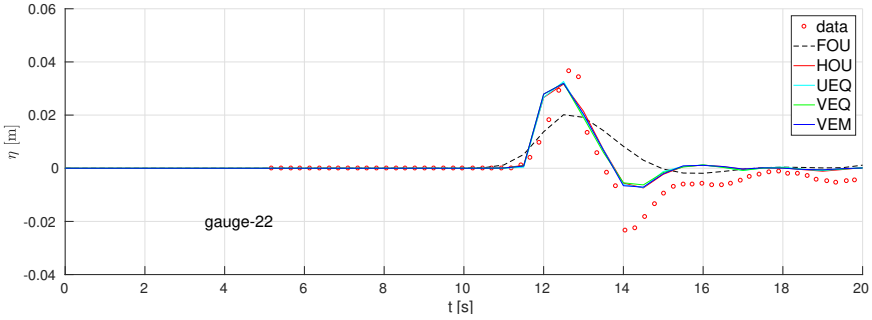
(a) Gauge 6



(b) Gauge 9

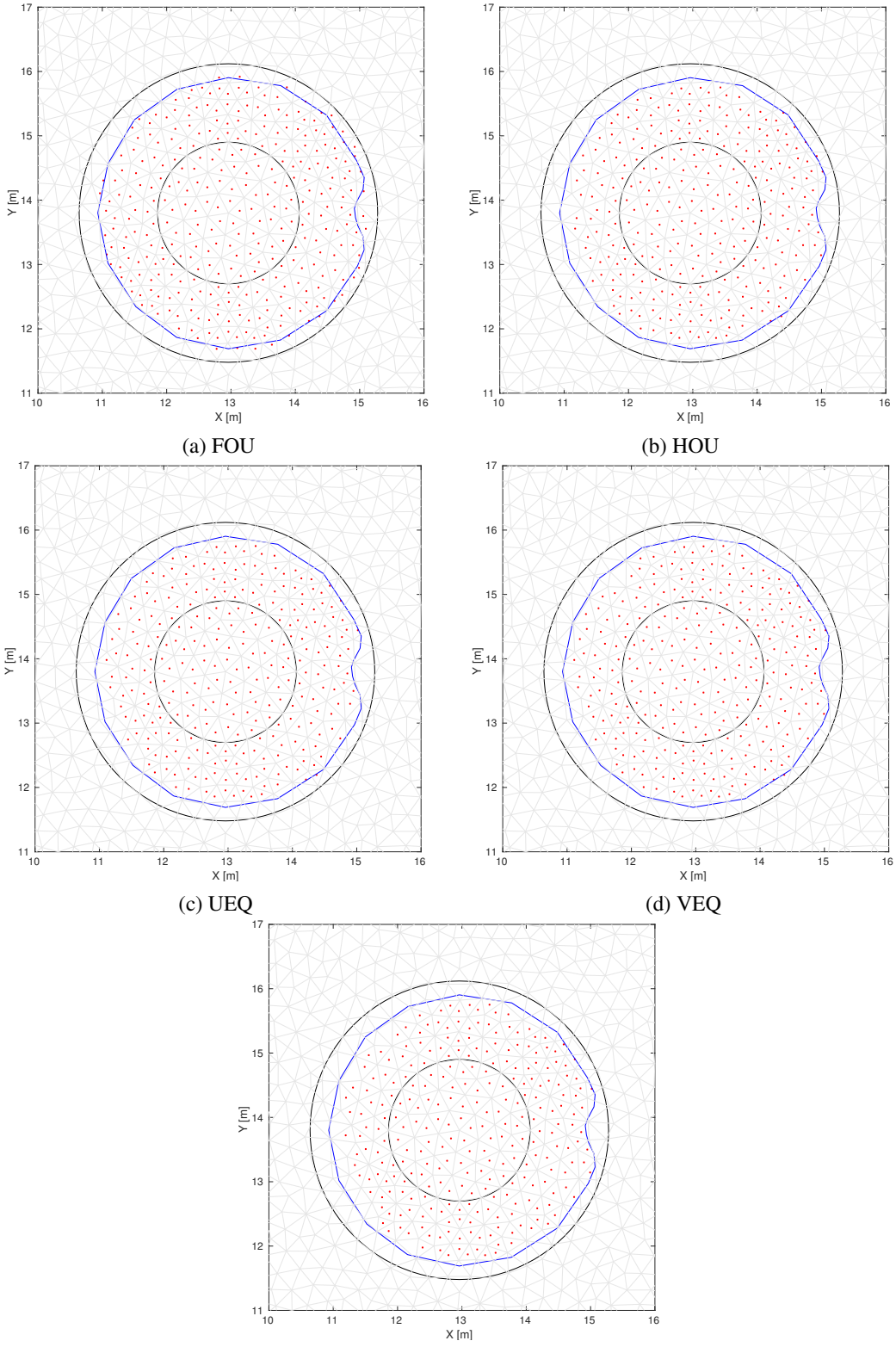


(c) Gauge 16



(d) Gauge 22

Figure 37. Tsunami wave impact on a conical island: time series of water level at 4 gauges.



(a) FOU

(b) HOU

(c) UEQ

(d) VEQ

(e) VEM

Figure 38. Tsunami wave impact on a conical island: measured max run-up (blue lines) and computed dry area (red dot).